

# An Improved Multi-label Classifier Chain Method for Automated Text Classification

Adeleke Abdullahi<sup>1</sup>

Noor Azah Samsudin<sup>2</sup>

Software Engineering Department  
Universiti Tun Hussein Onn  
Malaysia (UTHM)  
Batu Pahat, Malaysia

Shamsul Kamal Ahmad Khalid<sup>3</sup>

Information Security Department  
Universiti Tun Hussein Onn  
Malaysia (UTHM)  
Batu Pahat, Malaysia

Zuhaila Ali Othman<sup>4</sup>

Center for Artificial Intelligence  
Technology  
National University of Malaysia  
(UKM)  
Selangor, Malaysia

**Abstract**—Automated text classification is the task of grouping documents (text) automatically into categories from a predefined set. The conventional approach to classification involves mapping a single class label each to a data point (instance). In multi-label classification (MLC), the task is to develop models that could predict multiple class labels to a data instance. There exist several MLC methods such as classifier chain (CC) and binary relevance (BR). However, there are drawbacks with these methods such as random label sequence ordering issue. This study attempts to address this issue peculiar with the classifier chain method. In this paper, a hybrid heuristic evolutionary-based technique is proposed. The proposed PSOGCC is a combination of particle swarm optimization (PSO) and genetic algorithm (GA). Genetic operators of GA are integrated with the basic PSO algorithm for finding the global best solution representing an optimized label sequence order in the chain classifier. In the experiment, three MLC methods: BR, CC, and PSOGCC are implemented using five benchmark multi-label datasets and five standard evaluation metrics. The proposed PSOGCC method improved the predictive performance of the chain classifier by obtaining the best results of 98.66%, 99.5%, 99.16%, 99.33%, 0.0011 accuracy, precision, recall, *f1 Score*, and Hammingloss values, respectively.

**Keywords**—Text classification; multi-label classification; classifier chain; particle swarm optimization; genetic algorithm

## I. INTRODUCTION

Automated text classification (ATC) is the task of developing predictive models capable of categorizing text documents into distinct class labels from a predefined set. In other words, ATC is a technique that involves the process of managing and processing a vast number of documents in a continually increasing form. Conventionally, classification technique [1]–[3] focuses on the development of predictive model, a function that learnt to map an input  $x$  to an output  $y$ , *i. e.*,  $f: x \rightarrow y$ . This traditional approach to classification is otherwise termed single-label classification (SLC). Unlike the classical SLC technique, where an instance of a data sample is associated with a single class label, multi-label classification (MLC) [4]–[6] involves the problem of assigning to a data point (instance) multiple class labels simultaneously.

Given an input vector  $x = [x_1, x_2, \dots, x_n]^T$  and a vector of labels  $y = [y_1, y_2, \dots, y_k]^T$ , the goal of MLC is to build a model applicable in predicting one or more class labels

simultaneously provided the labels are not mutually exclusive. The multi-label classification concept primarily originated from text [5]. In a real-world scenario, a document (such as news article) could have multiple themes (topics) like entertainment, business, security, health, science, etc. To automate the categorization of such related textual data, MLC methods and techniques have been proposed. The existing MLC techniques could be broadly categorized into two approaches [6]: problem transformation and algorithm adaptation.

In problem transformation (PT) approach, the strategy involves transforming a multi-label problem into multiple single-label problems and learn one of the SLC algorithms (or classifiers) such as decision trees, for modeling the membership class (label). Subsequently, a new observation (test instance) is then predicted by combining the output of the positive predictions from the baseline classifiers. The PT strategy [7] is a very straightforward, easy, and flexible multi-label classification approach. Most of the conventional MLC algorithms such as binary relevance (BR), label powerset (LP), calibrated label ranking (CLR), and classifier chain (CC) adopt the PT strategy for MLC tasks.

Algorithm adaptation (AA) approach is based on inducing a conventional machine learning classification algorithm (single-label classifier) for multi-label problem. In other words, in AA strategy, a learning algorithm (classifier) such as support vector machine (SVM) is modeled and directly applied on MLC problems. This approach to MLC has been less applied by researchers due to its limitations such as lack of flexibility, complexity [8]. Notable algorithms that have adopted AA approach include ML- $k$ NN, BP-MLL, and BR- $k$ NN.

Classifier chain (CC) [9] [10] is one of the conventional MLC methods based on the problem transformation approach. The method is a direct extension of binary relevance (BR), developed to address the issue of label correlations. In BR, labels are taken as independent classifiers; hence the algorithm ignores labels inter-correlations. However, CC models consider labels as a chain-like structure, allowing communication (*i. e.*, sharing of predictions) among the underlying classifiers. The multi-label classification method has shown to be very competitive, achieving better classification results compared to other classical MLC methods such as BR [9].

Although, CC algorithm has been widely applied to several applications [11], [12]–[17], the method suffers from a major setback, which is the labels ordering issue [11], [12]. The conventional CC method adopts a random approach for labels sequence ordering, but studies have shown that the random labels sequence ordering may affect the performance of the classification method [11]. Attempts have been made to improve the original CC method, particularly to address the random labels sequence ordering issue, with several CC extensions proposed. This work attempts to further improve the standard CC method using a new alternative approach. In this paper, a hybrid heuristic evolutionary-based technique is proposed. The proposed PSOGCC optimization technique is a combination of particle swarm optimization (PSO) and genetic algorithm (GA).

The contributions of this work are grouped into three folds. First, we proposed an improved multi-label classifier chain method based on hybrid heuristic evolutionary techniques. Second, the proposed PSOGCC method is successfully demonstrated with standard benchmark multi-label datasets. Third, several conventional metrics are exhaustively employed to validate the performance of the proposed method against standard BR and CC methods in terms of Accuracy, Precision, Recall,  $f$ -Measure, and Hammingloss.

The rest of this paper is organized as follows. Section 2 reviewed related works, with focus on multi-label classifier chain method. Section 3 documented the experiment and method, and Section 4 presented the classification results. Section 5 concluded with direction to future works.

## II. RELATED WORKS

MLC is an emerging, growing field in the area of machine learning and data mining. MLC methods and techniques have been applied to various application domains including [4], [6], [18]–[20].

Specifically, there have been a growing number of works [11], [17], [21], [22] based on implementing and improving the multi-label classifier chain method. As aforementioned, CC is an extension of the classical BR method. The classifier chain method improved on BR by taking into consideration label correlations. The method works by modeling a set of binary classifiers (learning phase) based on the random label sequence ordering defined in the chain. The learning algorithm is then used to predict (a target label) taking into consideration the predictions of preceding labels in the chain. Given a new observation (prediction phase), the classifier makes prediction (following same procedure in the learning phase), by combining all positive predictions (outputs) of the classifiers. The performance of CC is sensitive to the label sequence order, which may be likely prone to “error propagation” in the chain. Several attempts have been made to overcome the limitations of CC.

In [23] an efficient label ordering approach was proposed for improving multi-label classifier chain accuracy. The proposed approach is based on exploiting semantic relationships among labels. The method achieved better

accuracy compared to the original CC method. Also, a decision function based on Bayesian network was proposed in [24] for multi-label classifier chain. Similarly, [22] employed the use of Bayesian network based on conditional entropy for discovering label correlation and order of labels in the chain classifier. The author in [25] proposed an improved classifier chain method based on conditional likelihood maximization. A  $k$  dependence classifier chains with label-specific function was developed. The method is shown to be effective. A cost-sensitive CC method was proposed in [12] for selecting low-cost features in multi-label classification. The method combined classifier chain with logistic regression dimensionality reduction technique.

In this paper, a hybrid heuristic evolutionary-based technique is proposed for improving the performance of multi-label classifier chain method. Heuristic techniques [26]–[30] are a set of intelligent self-learning algorithms developed to search for the optimum (best) solution to an optimization problem. Evolutionary-based heuristic methods are optimization algorithms that mimic the natural biological process (nature) in finding solutions to optimization problems. Most common and widely applied of the evolutionary-based optimization algorithms include: genetic algorithm, PSO, differential evolution, ant colony optimization algorithm, bee optimization algorithm, artificial immune system, cuckoo search, firefly algorithm, and tabu search algorithm.

The proposed technique applied in this work combined PSO and GA for finding the global solution that best represents an optimized label sequence order in the chain. Genetic operators: selection, crossover, and mutation, were integrated into the basic PSO algorithm for improving the search process, updating and maintaining diversity of the population (solutions). Details of the research methodology are presented in the next section.

### A. PSO Algorithm

Particle swarm optimization (PSO) is a population-based heuristic algorithm developed by Eberhart and Kennedy for solving optimization problems. The heuristic algorithm was influenced by the social behavior of species of animals such as birds flocking, fish schooling etc. In PSO algorithm (shown in Algorithm 1), a population entity called particle is assigned with position and velocity. A particle is a potential solution to a given problem. Each of the particles, represented as  $D$ -dimensional vector, moves around in the solution space, adjusting its position and velocity at every iteration using Eqn (1) and (2) respectively. Each particle has memory and remembers its previous best position  $pbest$  based on its experience. The global best represented as  $gbest$  is the collective best position in the swarm. Each particle knows the global best and move towards it. The performance of each particle (at every successive iteration) is measured using a fitness function.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1)$$

$$v_i(t+1) = wv_i(t) + c_1r_1[p_i(t) - x_i(t)] + c_2r_2[p_g(t) - x_i(t)] \quad (2)$$

where  $i = 1, 2, \dots, N$ ;  $t = 1, 2, \dots, T$ ;  $N$  represents swarm size and  $T$  is the maximum iteration limit;  $p_i$  and  $p_g$  are the local and global best solutions respectively;  $c_1$  and  $c_2$  are two acceleration constants in the value  $[0,1]$ ;  $r_1$  and  $r_2$  are two random numbers in the value  $[0,1]$ ;  $w$  is the inertia weight (which balances between local and global search);  $x_i(t)$  is the position of the particle and  $v_i(t)$  is the velocity of the particle at  $t$ -th iteration; particle ( $i^{th}$ ) position is denoted as  $x_i(t) = (x_{i1}, x_{i2}, \dots, x_{iD})$ , and velocity ( $i^{th}$ ) is denoted as  $v_i(t) = (v_{i1}, v_{i2}, \dots, v_{iD})$ .

Algorithm 1: Standard PSO Pseudocode

- Step 1: Initialize population of particles with random positions  $x$  and velocities  $v$ , swarm size  $s$
- Step 2: **For** each particle, let  $pbest = x$
- Step 3: calculate particles fitness  $f(x)$ ; update  $gbest$
- Step 4: **while** (termination criterion is not met)
- Step 5: **For**  $i = 1$  to  $S$
- Step 6: calculate the new velocity using Eq (3.1)
- Step 7: calculate the new position using Eq (3.2)
- Step 8: calculate  $f(x)$  of each particle
- Step 9: *if* ( $f(x) < f(pbest)$ )  $pbest = x$
- Step 10: *if* ( $f(pbest) < f(gbest)$ )  $gbest = pbest$
- Step 11: **end For**
- Step 12: **end For**
- Step 13: show the best solution found ( $gbest$ )

B. GA Algorithm

Genetic Algorithm (GA) is a global search optimization algorithm developed by Holland and based on the concept of natural selection adopted from the principle of Charles’ Darwin theory of evolution. GA is one of the most important and successful evolutionary-based heuristic method. The algorithm has been widely applied to several application problems [31]–[33]. The algorithm uses genetic operators: selection (or reproduction), crossover (or recombination), and mutation, to find (or produce) the global best solution to a given problem.

The evolutionary-based algorithm works (refer to Fig. 1) by first generate random initial population. At each generation, the quality of individuals (candidate solutions) is validated using a defined fitness function. Selection operator is applied to identify (select) individuals from the current generation based on the best fitness values. The process is improved through crossover and mutation operators until a new (better) population is created. The search ends with a termination criterion when the maximum iteration limit is reached or the best solution is found.

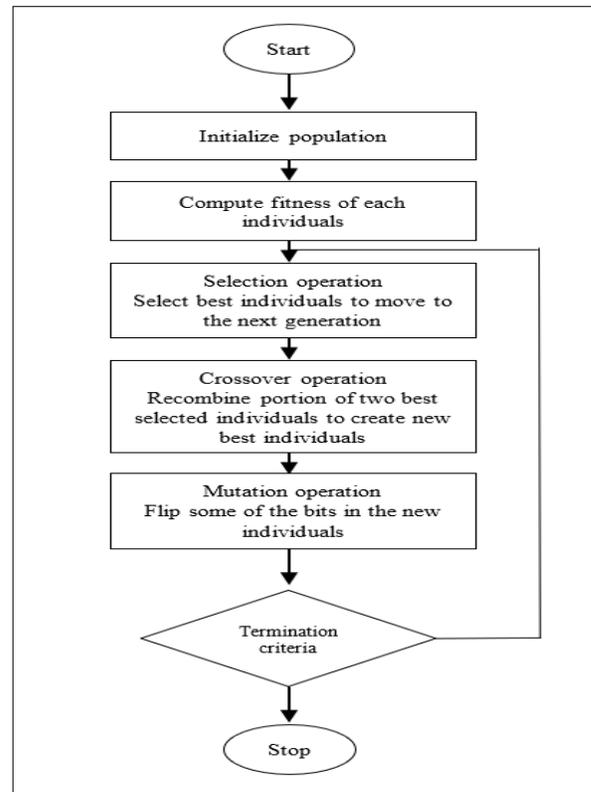


Fig. 1. Standard Genetic Algorithm.

III. METHODOLOGIES

In this paper, the experimental work comprises of four phases. These include input (data), preprocessing, classification, and output (results).

The experimental work is carried out using 5 benchmark multi-label datasets from Mulan (an open source library for multi-label classification problem). The standard datasets (in Table I) are from the most commonly experimented MLC datasets. The input data is preprocessed using StringToWordVector filtering tool and Term frequency inverse-document frequency (TFIDF). These are from the standard preprocessing techniques often applied in machine learning problems. The preprocessed data is stored in ARFF (Attribute-Relation File Format), the standard file format for machine learning using Mulan and Weka.

TABLE I. BENCHMARK ML DATASETS (WITH  $D$  = NO OF FEATURES;  $Q$  = NO OF CLASS LABELS;  $lc$  = LABEL CARDINALITY)

Dataset	Doman	#Instances	$D$	$Q$	$lc$
enron	Text	1702	1001	53	3.378
birds	Multimedia	645	260	19	1.014
flags	Image	194	19	7	3.392
genbase	Text	662	1186	27	1.252
yeast	Text	2417	103	14	4.237

### A. Proposed PSOGCC Multi-label Classification Method

The proposed MLC method is based on the concept of heuristic optimization technique, where the goal is finding the optimum solution to the search problem. The PSOGCC method (as shown in Fig. 2) is a hybrid of PSO and GA. The combined heuristic techniques are used to find the global best solution that best represents an optimized label sequence order in the chain classifier. PSO is an efficient, simple optimization algorithm and GA is a powerful, robust global search algorithm. Genetic operators: selection, crossover, and mutation, are applied for the population updates and reproduction of new generations (individuals).

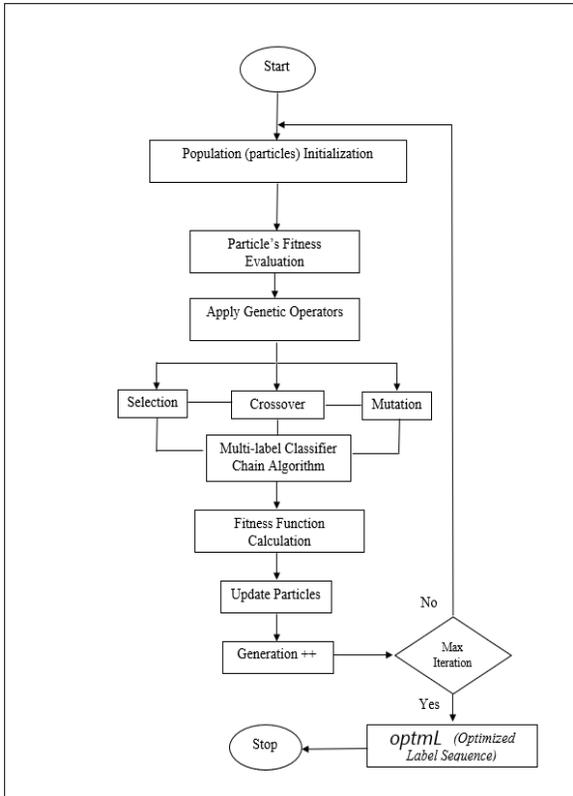


Fig. 2. Proposed PSOGCC Multi-label Classification Approach.

In PSOGCC (as shown in Algorithm 2), the optimization algorithm takes as input a training set  $T$  and produces as output an optimized label sequence  $optmL$ , representing the global optimum solution found in the chain. The entire algorithmic process could be broadly categorized into two: PSO loop (1 – 7) and GA loop (9 – 20).

In the first phase (*PSO loop*), population of particles (also called individuals) is initialized randomly with position  $x$ , velocity  $v$ , and swarm size  $s$ . Individual particles are represented as  $k$ - dimensional vectors (where  $k$  is equivalent to the number of predefined labels). The particles are encoded as integers representing label sequence indexes in the range value  $[1, q]$ . Individual particle's previous best position  $pbest$  is initialized with a copy of its current position  $x$ . The quality of particles is assessed using a defined fitness function  $f(x)$  in

Eq (3). Subsequently, the global best  $gbest$  is initialized with the index of the best fitted particle.

#### Algorithm 2: Pseudocode of the Proposed PSOGCC model

Input:  $T$  (training set)

Output:  $optmL$  (an optimized label sequence)

Step 1: Initialize population of particles (potential candidate solutions representing the label sequences) with random positions  $x$ , velocities  $v$ , and swarm size  $s$ . Set the particle's previous best position to the current position ( $pbest = x$ )

Step 2: Given a training set  $T$

Step 3: **For** all particles (label sequences) in the population **do**

Step 4: Build the classifier chain (CC) model (using standard **10 – fold** cross validation)

Step 5: Compute the particle's fitness  $f(x)$  using Eq (3)

Step 6: Update the population  $pbest$  and set the best particle  $gbest$  to the current population

Step 7: **end For**

Step 8: **repeat**

Step 9: Partition the training set  $T$  into **buildSet** and **validationSet**

Step 10: **For** all particles (candidate label sequence  $i$ ) in the current population **do**

Step 11: Construct the CC model using **buildSet** and label sequence  $i$

Step 12: Evaluate the fitness (quality) of the CC model using the **validationSet** and the fitness function  $g(x)$  defined in Eq (4)

Step 13: Apply Genetic operators:

Step 14: Using Tournament selection approach **SELECT** parents (particle with best fitness value)

Step 15: Generate new particles (child) from the old ones (parents) with **CROSSOVER** operator

Step 16: Apply **MUTATION** procedure (to the offspring)

Step 17: Update particles and population using **elitism age – based replacement** approach

Step 18: **end For**

Step 19: **Until (Max iterations (generations) reached)**

Step 20: return  $optmL$  (optimum solution rep labeled-ordered sequence)

In the second phase (*GA loop*), standard genetic operators: selection, crossover, and mutation, are applied. Classifier chain (CC) models are built and further evaluated using the fitness function  $g(x)$  defined in Eq (4). Genetic tournament selection strategy [34] is applied to select the best individuals to be recombined for producing a new generation (offspring). Order crossover operation [35] is performed using the selected individuals, resulting in the generation of new individuals. Thereafter, mutation operator is applied on the new individuals in order to avoid being trapped in the local minima. Age-based elitism replacement approach [36] is employed to replace the old generation with new ones while preserving a small group (elite individuals) in the population. This helps to improve and maintain diversity in the population. The PSOGCC implementation ends with the termination criteria and the global best solution  $optmL$  representing an optimized label sequence order in the chain classifier is returned.

$$Fitness f(x) = (\alpha * Acc) + (\beta * (\frac{N-T}{N})) \quad (3)$$

$$Fitness g(x) = \frac{(1 - (\frac{HL}{gMean})) + Acc + EM + F1}{N} \quad (4)$$

( $\alpha$  &  $\beta$ ) are control parameters (for balancing the trade-offs of particle's  $gbest$ ,  $gworst$ ,  $pbest$ , and  $pworst$ );  $Acc$  represents the accuracy of the baseline classifier; ( $N$  &  $T$ ) represent population size and neighborhood size, respectively.  $HL$ ,  $gMean$ ,  $Acc$ ,  $EM$ , and  $F1$  score are standard performance metrics.

#### IV. EXPERIMENTS AND RESULTS

This section details the experiments performed and the simulation results obtained. Five benchmark multi-label datasets with five conventional performance metrics were employed to validate the performance of the proposed PSOGCC method against the standard binary relevance (BR) and classifier chain (CC) multi-label classification algorithms. The classification results were compared in terms of: Accuracy ( $ACC$ ), Hammingloss ( $HL$ ), Precision ( $P$ ), Recall ( $R$ ), and  $f$ -Measure ( $F1$  score).

Accuracy ( $ACC$ ) [Eq 5] is a standard performance metric used to measure the correctly classified instances across data points. The higher the accuracy value, the better the classification algorithm. Precision [Eq 6], Recall [Eq 7], and  $f$ -Measure [Eq 8] are performance metrics often applied in classification problems to measure the degree of correctness of the positively classified instances. An effective classifier should have high precision, recall, and  $F1$  score. Lastly, Hammingloss [Eq 9] evaluation metric helps to measure the degree of incorrectness (misclassification) wrongly predicted by the classification algorithm. In general, a good classifier is one with high accuracy, precision, recall,  $F1$  score, and low Hammingloss values.

$$ACC = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (5)$$

$$P = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (6)$$

$$R = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap Z_i|}{|Y_i|} \quad (7)$$

$$F1 \text{ score} = \frac{1}{N} \sum_{i=1}^N \frac{2|Y_i \cap Z_i|}{(Y_i) + (Z_i)} \quad (8)$$

$$HL = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \Delta Z_i|}{k}, |Y_i \Delta Z_i| \text{ denotes symmetric diff btw } Y_i \text{ \& } Z_i \quad (9)$$

The three multi-label classification methods: PSOGCC, BR, and CC, produced competitive results. In Table II, the proposed PSOGCC achieved the highest accuracy result of 98.66% with the genbase multi-label dataset. Closely followed by the CC method with 98.15% accuracy while BR obtained 98.06%. From the accuracy results, it could be observed why the classifier chain (CC) outperformed the traditional BR algorithm. This is due to the limitation (associated with BR) of ignoring label correlations. Also, the proposed PSOGCC heuristic method outperformed the other two methods due to its combined advantages of considering label correlations and finding an optimized label sequence order, thereby addressing the limitation of the original CC method (i.e., random label sequence order in the chain).

Tables III to V presented the experimental results in terms of precision, recall, and  $f$ -Measure, respectively. Consistently, the proposed PSOGCC optimization algorithm outperformed both BR and CC multi-label methods. PSOGCC obtained the highest scores of 99.5%, 99.16%, and 99.33% precision, recall, and  $f1$  score respectively. These results further proved the effectiveness and superiority of the proposed method compared to the other two classical methods: binary relevance and classifier chain.

Finally, Table VI showed the Hammingloss values of the three classification methods obtained across the benchmark multi-label datasets. As aforementioned, Hammingloss metric helps to check the frequency of misclassification by the classifier. A good classifier should have less labels misclassified (i.e., low Hammingloss value). From the result, it could be seen that the proposed PSOGCC performed best compared to BR and CC. The method obtained the lowest Hammingloss value of 0.0011 with genbase dataset. The original CC method came second with 0.0102 Hammingloss value while BR performed the least (0.0121).

To further show a clearer and easier understanding of the classification results, the performance of the three MLC methods are presented in graphical forms as plotted in Fig. 3 to 7. The results comparisons showed the proposed PSOGCC had better performance across the multi-label datasets. This reflects the significance influence of finding an optimized label sequence order in the chain classifier.

TABLE II. CLASSIFICATION ( $ACC$ ) RESULTS OF PSOGCC, BR, AND CC

Datasets	Accuracy ( $ACC$ ) $\uparrow$		
	PSOGCC	BR	CC
enron	0.4046	0.3671	0.3671
birds	0.5515	0.5723	0.5725
flags	0.5586	0.5763	0.5700
genbase	0.9866	0.9806	0.9815
yeast	0.4537	0.4226	0.4219

TABLE III. CLASSIFICATION (P) RESULTS OF PSOGCC, BR, AND CC

Datasets	Precision (P) ↑		
	PSOGCC	BR	CC
enron	0.5959	0.6574	0.6576
birds	0.8153	0.8061	0.8064
flags	0.6741	0.6956	0.6943
genbase	0.9950	0.9947	0.9950
yeast	0.5796	0.5929	0.5950

TABLE IV. CLASSIFICATION (R) RESULTS OF PSOGCC, BR, AND CC

Datasets	Recall (R) ↑		
	PSOGCC	BR	CC
enron	0.4858	0.4481	0.4483
birds	0.6050	0.6403	0.6406
flags	0.6856	0.7741	0.7577
genbase	0.9916	0.9903	0.9908
yeast	0.6008	0.5613	0.5616

TABLE V. CLASSIFICATION (f1 Score) RESULTS OF PSOGCC, BR, AND CC

Datasets	f-Measure (f1 Score) ↑		
	PSOGCC	BR	CC
enron	0.5352	0.5329	0.5331
birds	0.6946	0.7137	0.7140
flags	0.6798	0.7328	0.7246
genbase	0.9933	0.9925	0.9928
yeast	0.5900	0.5767	0.5778

TABLE VI. CLASSIFICATION (HL) RESULTS OF PSOGCC, BR, AND CC

Datasets	Hammingloss (HL) ↓		
	PSOGCC	BR	CC
enron	0.0535	0.0540	0.0542
birds	0.0521	0.0515	0.0515
flags	0.2857	0.2747	0.2654
genbase	0.0011	0.0121	0.0102
yeast	0.2642	0.2588	0.2579

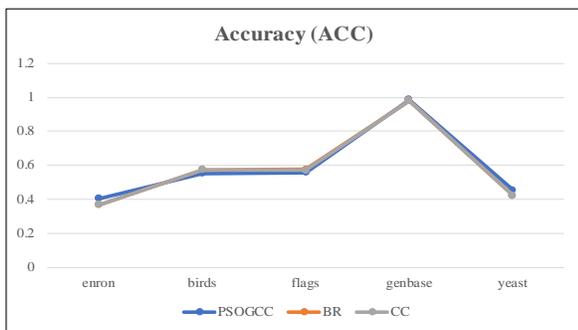


Fig. 3. Comparison of PSOGCC, BR, and CC in Terms of Accuracy.

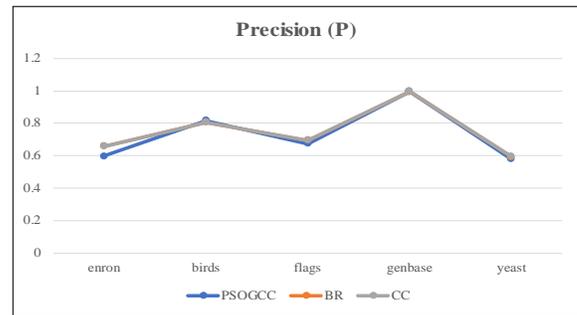


Fig. 4. Comparison of PSOGCC, BR, and CC in Terms of Precision.

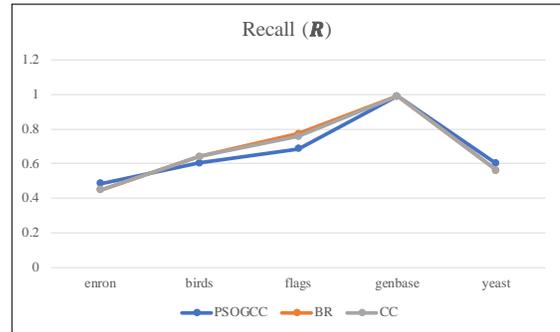


Fig. 5. Comparison of PSOGCC, BR, and CC in Terms of Recall.

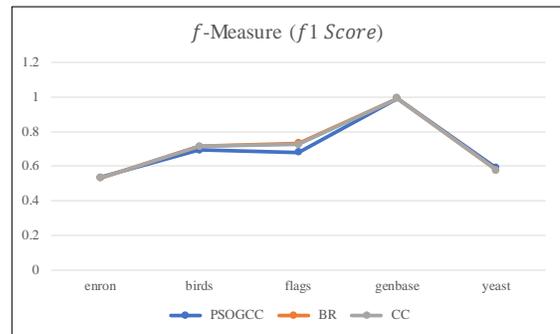


Fig. 6. Comparison of PSOGCC, BR, and CC in Terms of f1 Score.

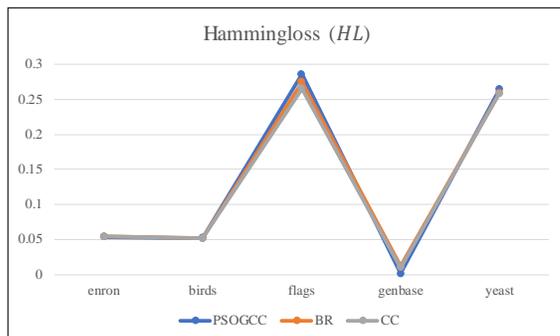


Fig. 7. Comparison of PSOGCC, BR, and CC in Terms of Hammingloss.

## V. CONCLUSION

Single-label classification (SLC) involves predicting a single class (output) for a particular data instance (input) whereas in multi-label classification (MLC), the task is to develop predictive models capable of assigning multiple class

labels simultaneously (to a single instance). In MLC, there are standard methods such as binary relevance (BR), classifier chain (CC), and label powerset (LP). There exist limitations with these methods such as ignoring label correlations (associated with BR), complexity (associated with LP), and random label ordering (associated with CC). This study attempted to improve the predictive performance of the multi-label CC method. In this work, the randomized label sequence order issue of CC is addressed. To achieve this, the study proposed a hybrid heuristic evolutionary-based technique.

Heuristic techniques involve developing a set of intelligent self-learning algorithms designed for finding the optimal best solution to an optimization problem. In this paper, PSOGCC multi-label classification method is proposed to extend the original CC method. The evolutionary-based algorithm is a combination of particle swarm optimization (PSO) and genetic algorithm (GA). The proposed PSOGCC method is used to find the global best solution representing an optimized label sequence order in the chain classifier. Genetic operators: selection, crossover, and mutation were integrated with the basic PSO for optimizing the search problem.

The experiment was conducted using five benchmark multi-label datasets. Furthermore, five evaluation metrics were applied to validate the performance (predictions) of the proposed PSOGCC against standard BR and CC methods. Results were presented in Tables II to VI in terms of accuracy, precision, recall,  $f$ -measure, and Hammingloss respectively. The proposed PSOGCC achieved the overall best classification results of 98.66%, 99.5%, 99.16%, 99.33%, 0.0011 accuracy, precision, recall,  $f$ -measure, and hammingloss values respectively.

In the future work, the proposed technique will be further validated using more multi-label datasets. Also, it is recommended to compare the performance of PSOGCC against other standard MLC algorithms. Finally, the research study will be further extended to employ other recent heuristic evolutionary-based techniques such as bat algorithm, whale optimization algorithm, and firefly algorithm etc.

#### ACKNOWLEDGMENT

The authors would like to thank the Ministry of Higher Education, Malaysia for supporting this research under Fundamental Research Grant Scheme Vot K213 (FRGS/1/2019/ICT02/UTHM/02/2) and Universiti Tun Hussein Onn Malaysia for Multidisciplinary Research, Vot H511.

#### REFERENCES

- [1] M. Iqbal, S. Ali, M. Abid, F. Majeed, and A. Ali, "Artificial Neural Network based Emotion Classification and Recognition from Speech," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 434–444, 2020.
- [2] A. K. Dehariya and P. Shukla, "Medical Data Classification using Fuzzy Main Max Neural Network Preceded by Feature Selection through Moth Flame Optimization," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 655–662, 2020.
- [3] A. Abdullahi, N. A. Samsudin, M. R. Ibrahim, M. S. Aripin, S. K. A. Khalid, and Z. A. Othman, "Towards IR4.0 implementation in e-manufacturing: Artificial intelligence application in steel plate fault detection," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 1, pp. 430–436, 2020.
- [4] M. Jethanandani, A. Sharma, T. Perumal, and J.-R. Chang, "Multi-label classification based ensemble learning for human activity recognition in smart home," *Internet of Things*, vol. 12, 2020.
- [5] A. Blanco, A. Casillas, A. Pérez, and A. Diaz de Ilarraza, "Multi-label clinical document classification: Impact of label-density," *Expert Syst. Appl.*, vol. 138, 2019.
- [6] Y. Xia, K. Chen, and Y. Yang, "Multi-label classification with weighted classifier selection and stacked ensemble," *Inf. Sci. (Ny)*, 2020.
- [7] M. Pushpa and S. Karpagavalli, "Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification," *Procedia Comput. Sci.*, vol. 115, pp. 572–579, 2017.
- [8] B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms," *Inf. Process. Manag.*, vol. 56, no. 1, pp. 212–227, 2019.
- [9] J. Read and L. Martino, "Probabilistic regressor chains with Monte Carlo methods," *Neurocomputing*, vol. 413, pp. 471–486, 2020.
- [10] P. Teisseyre, "Classifier chains for positive unlabelled multi-label learning," *Knowledge-Based Syst.*, vol. 213, 2021.
- [11] X. Jun, Y. Lu, Z. Lei, and D. Guolun, "Conditional entropy based classifier chains for multi-label classification," *Neurocomputing*, vol. 335, pp. 185–194, 2019.
- [12] P. Teisseyre, D. Zufferey, and M. Słomka, "Cost-sensitive classifier chains: Selecting low-cost features in multi-label classification," *Pattern Recognit.*, vol. 86, pp. 290–319, 2019.
- [13] F. Bellmann, L. Bunzel, C. Demus *et al.*, "Multi-Label Classification of Blurbs with SVM Classifier Chains," *Proc. 15th Conf. Nat. Lang. Process. (KONVENS 2019)*, pp. 293–299, 2019.
- [14] Z. Wang, T. Wang, B. Wan, and M. Han, "Partial classifier chains with feature selection by exploiting label correlation in multi-label classification," *Entropy*, vol. 22, no. 10, pp. 1–22, 2020.
- [15] W. Weng, D. H. Wang, C. L. Chen, J. Wen, and S. X. Wu, "Label Specific Features-Based Classifier Chains for Multi-Label Classification," *IEEE Access*, vol. 8, pp. 51265–51275, 2020.
- [16] Z. Yu, H. Hao, W. Zhang, and H. Dai, "A Classifier Chain Algorithm with K-means for Multi-label Classification on Clouds," *J. Signal Process. Syst.*, vol. 86, no. 2–3, pp. 337–346, 2017.
- [17] L. Sun and M. Kudo, "Optimization of classifier chains via conditional likelihood maximization," *Pattern Recognit.*, vol. 74, pp. 503–517, 2018.
- [18] R. Wang, S. Ye, K. Li, and S. Kwong, "Bayesian network based label correlation analysis for multi-label classifier chain," *arXiv*, vol. 554, pp. 256–275, 2019.
- [19] E. K. Y. Yapp, X. Li, W. F. Lu, and P. S. Tan, "Comparison of base classifiers for multi-label learning," *Neurocomputing*, vol. 394, pp. 51–60, 2020.
- [20] S. Nazmi, X. Yan, A. Homaifar, and E. Doucette, "Evolving multi-label classification rules by exploiting high-order label correlations," *Neurocomputing*, vol. 417, pp. 176–186, 2020.
- [21] T. T. Nguyen, M. T. Dang, A. V. Luong, A. W. C. Liew, T. Liang, and J. McCall, "Multi-label classification via incremental clustering on an evolving data stream," *Pattern Recognit.*, vol. 95, pp. 96–113, 2019.
- [22] R. Wang, R. Ridley, X. Su, W. Qu, and X. Dai, "A novel reasoning mechanism for multi-label text classification," *Inf. Process. Manag.*, vol. 58, no. 2, 2021.
- [23] B. Liu and G. Tsoumakas, "Dealing with class imbalance in classifier chains via random undersampling," *Knowledge-Based Syst.*, vol. 192, 2020.
- [24] T. Ali and S. Asghar, "Efficient Label Ordering for Improving Multi-label Classifier Chain Accuracy," *J. Natn. Sci. Foundation Sri Lanka*, vol. 47, no. 2, pp. 175–184, 2019.
- [25] G. Varando, C. Bielza, and P. Larrañaga, "Decision functions for chain classifiers based on Bayesian networks for multi-label classification," *Int. J. Approx. Reason.*, vol. 68, pp. 164–178, 2016.
- [26] X. Chu *et al.*, "An artificial bee colony algorithm with adaptive heterogeneous competition for global optimization problems," *Appl. Soft Comput. J.*, vol. 93, 2020.

- [27] H. S. Alhadawi, D. Lambić, M. F. Zolkipli, and M. Ahmad, "Globalized firefly algorithm and chaos for designing substitution box," *J. Inf. Secur. Appl.*, vol. 55, 2020.
- [28] H. Wang, X. Lang, and W. Mao, "Voyage optimization combining genetic algorithm and dynamic programming for fuel/emissions reduction," *Transp. Res. Part D Transp. Environ.*, vol. 90, 2020.
- [29] N. Taheranpour and S. Talebi, "Development of a new efficient method using genetic algorithm for increasing of fuel rod life time," *Prog. Nucl. Energy*, vol. 131, 2020.
- [30] J. Liu, X. Ma, X. Li, M. Liu, T. Shi, and P. Li, "Random convergence analysis of particle swarm optimization algorithm with time - varying attractor," *Swarm Evol. Comput.*, vol. 61, 2020.
- [31] E. T. Bacalhau, L. Casacio, and A. T. de Azevedo, "New hybrid genetic algorithms to solve dynamic berth allocation problem," *Expert Syst. Appl.*, 2020.
- [32] A. Santiago, B. Dorronsoro, H. J. Fraire, and P. Ruiz, "Micro-Genetic algorithm with fuzzy selection of operators for multi-Objective optimization:  $\mu$ FAME," *Swarm Evol. Comput.*, vol. 61, 2019.
- [33] A. Singh and A. Khamparia, "A hybrid whale optimization-differential evolution and genetic algorithm based approach to solve unit commitment scheduling problem: WODEGA," *Sustain. Comput. Informatics Syst.*, vol. 28, 2020.
- [34] Z. Li, J. Huang, and M. Ding, "Comparison and analysis of different selection strategies of genetic algorithms for fuel reloading optimization of Thorium-based HTGRs," *Nucl. Eng. Des.*, vol. 373, 2020.
- [35] B. Koohestani, "A crossover operator for improving the efficiency of permutation-based genetic algorithms," *Expert Syst. Appl.*, vol. 15, 2020.
- [36] S. Rani, B. Suri, and R. Goyal, "On the effectiveness of using elitist genetic algorithm in mutation testing," *Symmetry (Basel)*, vol. 11, no. 9, 2019.