A New Feature Engineering Approach for Multi-label Classification using Feature Encoding and Soft-loss

Worawith Sangkatip

A Thesis Submitted in Partial Fulfillment of Requirements for

degree of Doctor of Philosophy in Information Technology

May 2022

วิธีการวิศวกรรมการแทนข้อมูลด้วยคุณลักษณะใหม่สำหรับการจำแนกประเภทแบบหลายเลเบล โดยใช้การเข้ารหัสคุณลักษณะและซอฟต์ลอส

วิทยานิพนธ์

ของ

วรวิทย์ สังฆทิพย์

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

พฤษภาคม 2565

A New Feature Engineering Approach for Multi-label Classification using Feature Encoding and Soft-loss

Worawith Sangkatip

A Thesis Submitted in Partial Fulfillment of Requirements

for Doctor of Philosophy (Information Technology)

May 2022

The examining committee has unanimously approved this Thesis, submitted by Mr. Worawith Sangkatip , as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology at Mahasarakham University

Examining Committee

.....Chairman
(Asst. Prof. Wararat Songpan , Ph.D.)

.....Advisor
(Asst. Prof. Phatthanaphong Chompoowises , Ph.D.)

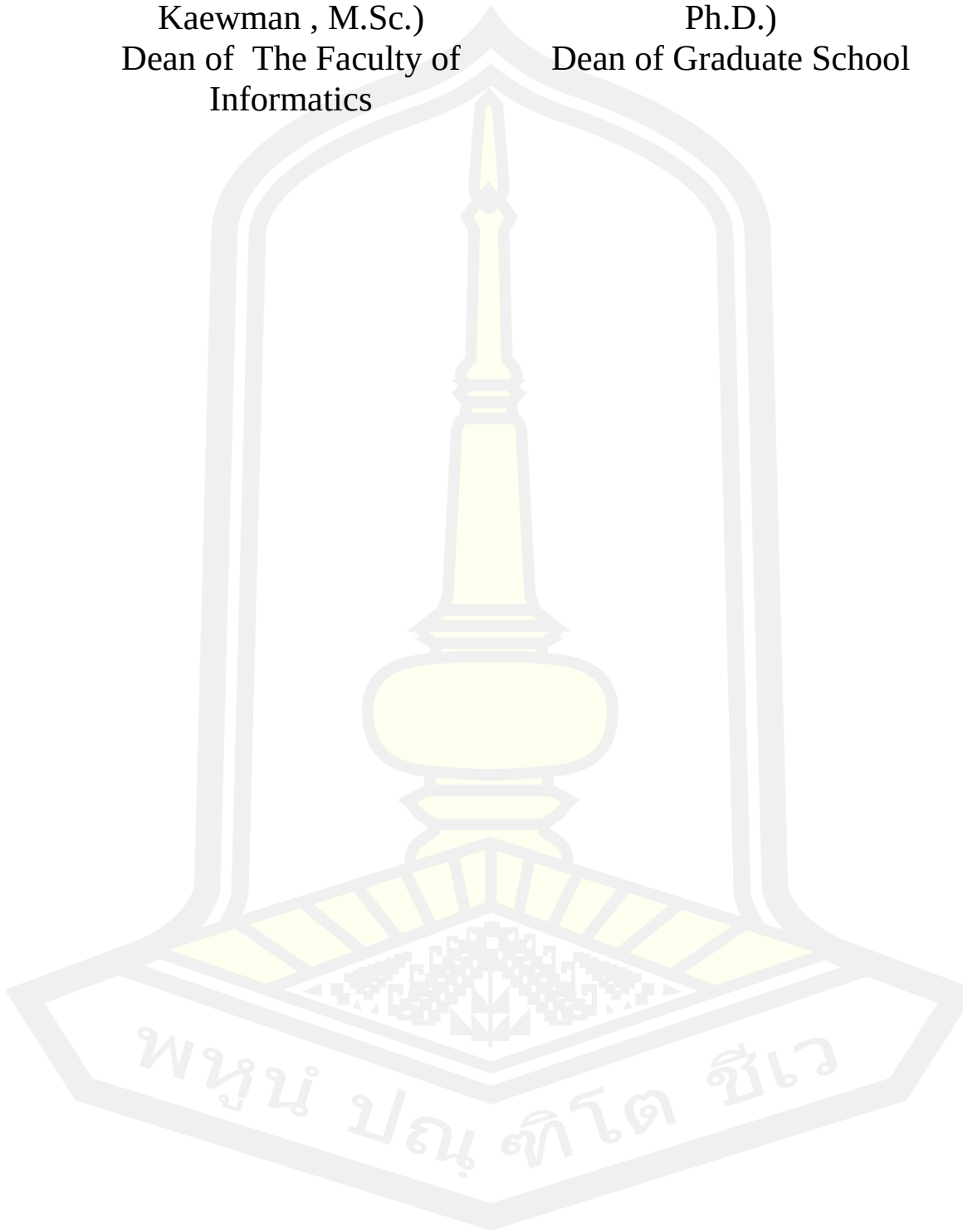
.....Committee
(Asst. Prof. Chatklaw Jareanpon , Ph.D.)

.....Committee
(Asst. Prof. Olarik Surinta , Ph.D.)

.....Committee
(Asst. Prof. Rapeeporn Chamchong , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology

.....	.....
(Asst. Prof. Sasitorn Kaewman , M.Sc.) Dean of The Faculty of Informatics	(Assoc. Prof. Krit Chaimoon , Ph.D.) Dean of Graduate School



TITLE A New Feature Engineering Approach for Multi-label Classification using Feature Encoding and Soft-loss

AUTHOR Worawith Sangkatip

ADVISORS Assistant Professor Phatthanaphong Chompoowises , Ph.D.

DEGREE Doctor of Philosophy **MAJOR** Information Technology

UNIVERSITY Mahasarakham University **YEAR** 2022

ABSTRACT

This thesis aims to improve the performance of multi-label classification (MLC) potentially. The research objectives are to improve the MLC performance using feature encoding and Soft-loss. This work attempts to drive three research questions and investigate scientific approaches to respond to the questions to achieve the research objectives. The thesis's contribution is divided into three folds : (i) Results of comparing state-of-the-art MLC methods with the non-communicable disease dataset. (ii) Feature reconstruction technique using an AutoEncoder network that encodes the features and labels, which improves the efficiency of MLC on the standard dataset. (iii) Applying the label patterns of the data to improve the classification performance.

Keyword : Multi-Label Classification, Feature Reconstruction, Label Correlation, Artificial Neural Network

ACKNOWLEDGEMENTS

I want to express thanks to the thesis advisor, Assistant Professor Dr. Phatthanaphong Chompoowises, for his invaluable help and constant encouragement throughout this research. This thesis would not have been completed without all the support I have always received from him. I am most grateful for his teaching and advice.

Thank you to my committee members, Assistant Professor Dr.Wararat Songpan, Assistant Professor Dr.Chatklaw Jareanpon, Assistant Professor Dr.Olarik Surinta, and Assistant Professor Dr.Rapeeporn Chamchong. Your encouraging words and thoughtful, detailed feedback have been very important to me.

I would like to thank Assistant Professor Dr. Jiratta Phuboon-Ob for his assistance and support for everything. And encourage them to do the thesis.

This research project was financially supported by Mahasarakham University. I would like to thank you for funding this research. Moreover, thank you, Suddhavej Hospital, Faculty of Medicine, Mahasarakham University, for providing information for use in research.

Thank you to Rajabhat Maha Sarakham University for providing opportunities and supporting scholarships for Ph.D. study.

I most gratefully acknowledge my parents and my friends for all their support throughout the period of this research. Also, I extend my thanks to all my colleagues at Major of Information Technology, Rajabhat Maha Sarakham University, for their continuous encouragement and support.

Finally, I most gratefully acknowledge my family for their encouragement and support throughout my studies.

Worawith Sangkatip

TABLE OF CONTENTS

Page

ABSTRACT ..... D

ACKNOWLEDGEMENTS ..... E

TABLE OF CONTENTS ..... G

List of Tables ..... J

List of Figures L

Chapter 1 1

Introduction 1

1.1 Introduction 1

1.2 Research Questions: RQ 4

1.3 Objectives 5

1.4 Contributions 5

1.5 Framework of the Proposed Method 7

Chapter 2 9

Non-Communicable Diseases Classification using Multi-Label Learning Techniques 9

2.1 Introduction 9

2.2 Related work 10

2.3 Methodology 11

2.3.1 Multi-label classification methods 12

2.3.2 Dataset 12

2.3.3 Data Preprocessing 14

2.3.4 Evaluation Measures 16

2.4 Experimental 17

2.4.1 Experimental Setup 17

2.4.2 Experimental Result 17

2.5 Conclusion 19

Chapter 3 20
Improving Multi-label Classification using Feature Reconstruction Methods 20
3.1 Introduction 20
3.2 Multi-label Classification 23
3.2.1 Transformation-Based Classifiers 24
3.2.2 Adaptation-Based Classifiers 25
3.2.3 Ensemble-Based Classifiers 26
3.3 Methodology 27
3.3.1 Dataset 27
3.3.2 Feature Reconstruction using AutoEncoder 28
3.4 Experiment Setup and Evaluation Metrics 31
3.5 Results and Discussion 34
3.6 Conclusion 47
Chapter 4 49
A Comparative Study of Applying Neural Network-based Techniques for Solving Multi-label Classification Problems 49
4.1 Introduction 49
4.2 Neural Network for Multi-label Classification 52
4.2.1 Preliminaries 52
4.2.2 Constructing Neural Network for the Classification 52
4.2.3 Backpropagation for Multi-Label Learning (BP-MLL) 53
4.3 Materials and Methods 54
4.3.1 Dataset Pattern Analysis 55
4.3.2 Method 58
4.4 Experiments and Results 61
4.4.1 Experiment setup 61
4.4.2 Experiment Results 63
4.5 Discussion 71
4.6 Conclusion 72

Chapter 5 73
Discussion 73
5.1. Answers to the Research Questions 74
5.2. Future Work 77
REFERENCES 78
REFERENCES 87
BIOGRAPHY 88

List of Tables

Page

Table 1 Multi-Label Physical Examination Records Description. 13
Table 2 Equation 1 ICD-10 Code for the four NCDs. 14
Table 3 The Detail Information of NCDs Dataset. 15
Table 4 Summarize the NCDs Dataset used in the Experiment. 15
Table 5 Performance evaluation for different multi-label methods. 18
Table 6 Representation of data instance. 23
Table 7 Example of label powerset multi-label transformation. 24
Table 8 The transformed problem becomes a 3-class problem. 25
Table 9 Example of problems subsets in RAkEL technique. 27
Table 10 Multi-label Datasets. 28
Table 11 Comparative results for Yeast dataset. 35
Table 12 Comparative results for Emotions dataset. 36
Table 13 Comparative results for Scene dataset. 37
Table 14 Comparative results for Medical dataset. 38
Table 15 Comparative results for Cal500 dataset. 39
Table 16 Comparative results for Birds dataset. 40
Table 17 Comparative results for Enron dataset. 41
Table 18 Comparative results for Foodtruck dataset. 42
Table 19 Comparative results for Yeast dataset between Native and TEN. 43
Table 20 The details of multi-label datasets used in this work. 55
Table 21 The number of label patterns of the training dataset used in this work. 56
Table 22 Pattern of the predicted test dataset. 58
Table 23 The performance of multi-label algorithm in terms of Precision ↑. 63
Table 24 The performance of multi-label algorithm in terms of Recall ↑. 64
Table 25 The performance of multi-label algorithm in terms of F1 ↑. 64
Table 26 The performance of multi-label algorithm in terms of Macro Precision ↑. 65

Table 27 The performance of multi-label algorithm in terms of Macro Recall ↑.....65
Table 28 The performance of multi-label algorithm in terms of Macro F1 ↑.....66
Table 29 The performance of multi-label algorithm in terms of Micro Precision ↑....66
Table 30 The performance of multi-label algorithm in terms of Micro Recall ↑. 67
Table 31 The performance of multi-label algorithm in terms of Micro F1 ↑. 67
Table 32 The performance of multi-label algorithm in terms of Hamming Loss ↓.....68

List of Figures

Page

Figure 1 The overall framework presented to improve MLC in the thesis. 7
Figure 2 The multi-label classification methods used in this research. 11
Figure 3 Database relationship of medical and health data standards. 13
Figure 4 Performance accuracy comparison with multi-label classification methods. 18
Figure 5 Performance hamming loss comparison with multi-label classification methods. 19
Figure 6 The overall process of the feature reconstructing for solving MLC. 29
Figure 7 AutoEncoder architecture EN. 29
Figure 8 AutoEncoder architecture TEN. 30
Figure 9 The results were obtained from the proposed feature reconstruction method and the native data feature. 44
Figure 10 The BR measurement results compare the native feature with TEN. 45
Figure 11 The CC measurement results compare the native feature with TEN. 45
Figure 12 The LP measurement results compare the native feature with TEN. 45
Figure 13 The MLTSVM measurement results compare the native feature with TEN. 46
Figure 14 The ML-KNN measurement results compare the native feature with TEN. 46
Figure 15 The RAkELd measurement results compare the native feature with TEN. 46
Figure 16 The overall process of the proposed method. 54
Figure 17 The distribution of the label patterns in each of the datasets. 57
Figure 18 Illustration of the visualization of the loss function of the datasets. 69
Figure 19 Comparison of proposed against other methods using Bonferroni-Dunn statistic (CD=1.974, $\alpha = 0.05$) 70

Chapter 1

Introduction

1.1 Introduction

Multi-label classification (MLC) is part of supervised machine learning (M. L. Zhang & Zhou, 2014) (Herrera, Charte, Rivera, & Del Jesus, 2016). The problem is one of the classification problems that has gained extensive attention in the research in machine learning (Mencía et al., 2018). In general, real-world applications usually contain more than one data entity or classes; such an image may contain different tangible objects in one single image scene. In the medical domain, classifying patients with multiple diseases can be one of the applications of applying the MCL technique (Sangkatip & Phuboon-Ob, 2020). Images detection with multiple objects; video clips are several categories; classification of sounds in different emotions undergoes applications that use MCL. From the information perspective, traditional classification techniques with only one single class may not be applicable to solve MCL problems. Therefore, MLC methods have been specifically introduced to solve these complex problems (more than one data entity). The MLC method was presented in 2004 by Boutell et al. (2004). The work aimed to present the classification of objects and, yet, to detect multi-objects in an image. From then on, there began to be more severe research that proposed a solution to the problem of multi-label classification. Furthermore, there is a standardized data set in many domains published for researchers to experiment (Tsoumakas, Spyromitros-Xioufis, et al., 2011).

Tsoumakas & Katakis. (2007a) divided MLC techniques into two groups: Adaptation Methods (AM) and Problem Transformation Methods (PTM). The first group, the AM method, is a method that improves the algorithm on the classification. For example, the Decision Tree algorithm C4.5 is implemented to enable multi-label classification, called ML-C4.5 (Clare & King, 2001), and the K-Nearest Neighbors algorithm is modified. MLC is possible, known as Multi-label K-Nearest Neighbor

(ML-*K*NN) (M. L. Zhang & Zhou, 2007). Moreover, the second group, the PTM method, is the method for converting problems from multiple labels into a single class first so that the conventional classification algorithm can classify multiple labels. Example, Binary Relevance (BR) (Tsoumakas & Katakis, 2007b), Classifier Chains (CC) (Read et al., 2009), Label Power-set (LP) (Tsoumakas et al., 2008). Madjarov et al. (2012a) presented an experiment to compare the effectiveness of a MLC method. The MLC method was divided into three groups,s retaining the same AM and PTM methods. A new group called Ensemble Methods (EM) was added, which was developed by combining PTM methods to improve classification efficiency. Such as RAndom k-labELsets (RAkEL) (Tsoumakas & Vlahavas, 2007b), Ensemble of Pruned Sets (EPS) (Read et al., 2008), and Ensemble of Classifier Chains (ECC) (Read et al., 2009).

The current MLC challenges are focused on improving classification efficiency with greater accuracy. More research has been done to analyze feature-label relationships. Feature Engineering (FE) (Guozhu & Huan, 2018; Hafeez et al., 2021) is divided into several tasks, for example, Feature Selection (FS), Feature Transformation (FT), and Feature Reconstruction (FR). Dimensional reduction is one of the techniques used to transform data features. There are two categories of dimensionality reduction methods. One is feature selection and feature transformation. Method FS keeps only useful features and dismisses others, while FT constructs a new but smaller number of features out of the original ones (Deng et al., 2013b). The current FT method can be applied by implementing, for example, deep learning algorithms (Patterson & Gibson, 2017) and unsupervised network algorithms, which learn to encode data to extract the relationships of the data. Y. Cheng et al. (2019) used a deep learning technique to build and extract relationships between attributes and labels in a multi-label classification. Feature reconstruction, a transformation process, can be considered a tool to generate a set of new feature sets (based on the original data features). The reconstructed features are anticipated to be compact and descriptive, which can be used in the classification process.

There is a very successful method for increasing the efficiency of multi-label classifications, known as Label Correlation (LC) (Fan et al., 2021; J. Li et al., 2022; Nazmi et al., 2021). It is a method that cares about label relationships and believes

that the resulting labels are interdependent. And are not independent of each other. Therefore, it is proposed to find the probabilities of a label from the relation of the data to the label. M.-L. Zhang. (2011) proposed that LIFT used 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 established by creating additional attributes of the data (Gao et al., 2020). Huang et al. (2018) proposed a technique to learn the dispersion of label attributes, including common attributes. They applied double-label correlation to differentiate labels for each category.

The literature review from its inception to the current research of the MLC, a method, is still in the spotlight. Furthermore, there are still challenges for researchers today. Interests in tackling current MLC challenges can be grouped into several groups. The first group is interested in improving MLC's existing algorithms, such as PTM, AM, and EM types, to be more efficient. The second group, which optimizes the dimensions of the instance data, reduces the dimension of the features and shrinks the label to a smaller dimension for a more straightforward classification. And the third group analyzes the correlation of features with the labels or the correlation between the labels that occur the most. All three groups are still in the interest of researchers to improve the efficiency of multi-label classifications with greater accuracy.

This thesis proposed a method to improve the efficiency of MLC to achieve a more accurate multi-class classification. The work is divided into three tasks: First, this was to use the non-communicable chronic disease diagnosis dataset. Experiment with popular traditional MLC methods. We compare the efficiency of the classification of each method. Second, this task proposes a method for establishing a relationship between a feature and a label to optimize classification by presenting a feature reconstruction method with an AutoEncoder algorithm to learn and create new features that are related to labels and then apply the new features to experiment with MLC methods and compare their performance. Third, this task proposes a technique that performs that classification by integrating the information of label patterns exhibited in the data. This information is anticipated to be helpful in training an

Artificial Neural Network (ANN) to obtain a generalized model and predict outcomes to the existing label patterns in data.

1.2 Research Questions: RQ

RQ1: The MLC methods have been proposed to solve the problem of classifying more than one class, also known as multi-label, for more efficient classification. The initial or traditional method is presented in several groups as AM, PTM, and EM. These methods are referenced and are the base for developing new methods such as BR, CC, LP, and ML-*K*NN. They provide excellent performance in multi-label classification. Therefore, this research is interested in applying the popular traditional method of MLC. It uses the Non-Communicable Disease (NCDs) diagnosis dataset to experiment and collect data from Suthavej Hospital. It is information on patients with chronic non-communicable diseases. For example, patients with diabetes often have hypertension. From the information, the patient data had more than one concomitant diagnosis. The MLC approach is needed to classify multiple diseases together for the above problem. The dataset is introduced into the MLC process using the defined methods. The results were compared to measure the efficiency of the classification of each method. Furthermore, the conclusion is which method can most accurately classify data for diagnosing chronic non-communicable diseases.

RQ2: The MLC performance improvements used features jointly with labels are well-known and allow for higher classification efficiency (Fan et al., 2021; J. Li et al., 2022; Nazmi et al., 2021). This work proposes a method for reconstructing features from learning the relationship between features and labels because features are an essential factor in classifying data in which labels use the AutoEncoder algorithm to learn to correlate features with labels. It will get a new feature that has changed the dimensions of the data may be increased or reduced. Nevertheless, the relationship of the feature data is more indicative of the label. This method permits the MLC algorithm to classify more accurately.

RQ3: One of the possible solutions for performing the MLC task is to investigate the patterns of class labels in the dataset. The classification can be carried out in multi-class classification family schemes. Power subset is a general technique

that converts MLC to multi-class problems. Based on the same principles, this research question will investigate the drive into using the pattern of a label (or data classes) in the data to assist the classification. Would the patterns of labels be used in training a model to obtain a generalized model for MLC.

1.3 Objectives

Improve the multi-label classification performance using feature encoding and Soft-loss.

1.4 Contributions

This thesis aims to propose methods to improve the efficiency of MLC by adapting existing methods to achieve better-generalized MLC. This thesis uses generic real-world datasets comprising the Non-Communicable Diseases (NCDs). Dataset collected from Suthavej Hospital and used more than eight standard datasets available for the research. The experiments use both the NCDs and the standard datasets. The traditional MLC methods are implemented to evaluate the effectiveness of the classification methods. Furthermore, this work improves the efficiency of MLC, integrating the relationship between features and labels of the data. The contribution and the organization of the thesis be described as follows:

In Chapter 2, this thesis investigates the traditional comparative methods of multi-label classification. The major used four methods are Binary relevance (BR), Classifier Chains (CC), The RAndom k-labELsets (RAkEL), and Multi-label K-Nearest Neighbor (ML-*K*NN). The work in this chapter used a non-communicable chronic disease dataset from Suthavej Hospital to generate the classification models and construct the evaluation experiments. The results obtained from the experiments were compared to investigate the empirical significance of the performance of different classification methods. The work was published at the 5th International Conference on Information Technology: InCIT2020. (*Sangkatip, W., & Phuboon-Ob, J. (2020). Non-Communicable Diseases Classification using Multi-Label Learning Techniques. International Conference on Information Technology (InCIT), 17–21).*

In Chapter 3, this thesis proposed a method to improve the MLC performance using the feature reconstruction method. The work applies an AutoEncoder network to capture the relationship between data features and their labels, called Target-label to the auto-ENcoder network (TEN). The work evaluates the performance of the technique used eight standard datasets in the experiments with new features. The proficiency of the generated features, six MLC methods as BR, CC, LP, MLTSVM, ML-*K*NN, and RAkELd were applied. Ten evaluation metrics were utilized, i.e., Precision, Recall, F1, Hamming Loss, Micro Precision, Micro Recall, Micro F1, Macro Precision, Macro Recall, and Macro F1. The experimental results demonstrate that the proposed technique provided promising results. Thus, the work was published and presented in the Asia Joint Conference on Computing: AJCC2022, which was published in the journal Current Applied Science and Technology, entitled: "*Improving Multi-label Classification using Feature Reconstruction Method*s.".

Chapter 4 presents a technique to perform MLC using ANN techniques. The proposed method aims at improving the classification performance by investing in an optimization constraint for better-generalized models. The work examines an alternative approach to the loss function used during training. In this work, the patterns of the multi-label class are explored. These patterns can essentially be used to construct a trained network that encourages the training to converge toward the existing patterns in the train data. The pattern information will be used to implement additional loss terms, so-called Soft-loss. It will be weighted and tuned toward the optimal solution. The Soft-loss anticipated the training to converge and direct the solution to the existing patterns in the data. The Soft-loss has been divided into two classification techniques, i.e., (i) patterns of the label (POL) and (ii) label similarity (LSIM). This method permits the MLC more accurately than other methods.

1.5 Framework of the Proposed Method

This thesis aims to propose methods to improve the efficiency of MLC by adapting methods to achieve better MLC. It is divided into three tasks corresponding to research questions, as demonstrated in Figure 1.

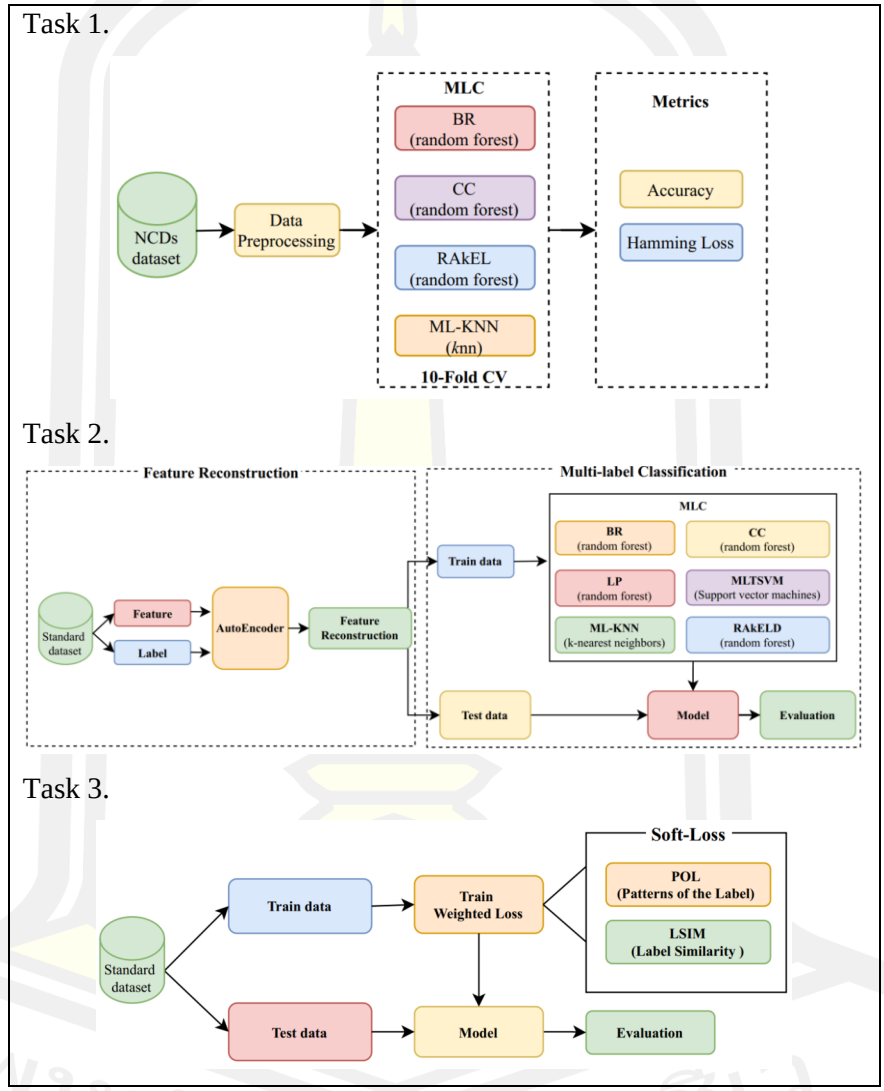


Figure 1 The overall framework presented to improve MLC in the thesis.

In task 1, this work investigates the traditional comparative methods of MLC. We used four MLC methods as BR, CC, RAkEL, and ML-*K*NN. The work used an NCDs dataset to generate the classification models and construct the evaluation experiments. The results obtained from the experiments and compared to investigate

the empirical significance of the performance of different classification methods, as the work details in chapter 2.

In task 2, this work proposed a method to improve the MLC performance using the feature reconstruction method. The work applies an AutoEncoder network to capture the relationship between data features and their labels, called Target-label to the auto-ENcoder network (TEN). The work evaluates the performance of the technique used eight standard datasets in the experiments with new features. The proficiency of the generated features, six MLC methods as BR, CC, LP, MLTSVM, ML-*K*NN, and RAkELd were applied. The experimental results demonstrate that the proposed technique provided promising results, as the work details in chapter 3.

In task 3, this work proposed a method to perform MLC using ANN techniques. The proposed method aims at improving the classification performance by investing in an optimization constraint for better-generalized models. The work examines an alternative approach to the loss function used during training. In this work, the patterns of the multi-label class are explored. These patterns can essentially be used to construct a trained network that encourages the training to converge toward the existing patterns in the train data. The pattern information will be used to implement additional Soft-loss. The Soft-loss has been divided into two classification techniques, i.e., (i) patterns of the label (POL) and (ii) label similarity (LSIM). This method permits the MLC more accurately than other methods, as the work details in chapter 4.

Chapter 2

Non-Communicable Diseases Classification using Multi-Label Learning Techniques

2.1 Introduction

Non-communicable diseases (NCDs) are a type of illness that is not particularly caused by bacteria or viruses. NCDs are not transmissible directly from one person to another; however, they can potentially affect persons who undergo NCDs' habits or lifestyles. The World Health Organization (WHO) (WHO, 2021) reports that NCDs cause 41 million deaths each year, accounting for 71% of deaths in the world. The most cause of death is cardiovascular disease, cancer, respiratory disease, and diabetes consequently. In Thailand, NCDs cause several deaths for both males and females, especially for persons who are over 30 years old (International Health Policy Program, 2015). From the studies, NCDs are stimulated and result from various factors: drinking alcohol, smoking, not exercising, eating much sweet and salty food, and stress.

Data obtained from the empirical experiments from the laboratory and disease diagnosis shows that NCD patients always have multi-morbidities. For example, diabetic patients usually have hypertension symptoms. Several clinical processes are carried out in the diagnosis procedures, e.g., principal diagnosis, comorbidity diagnosis, complication diagnosis, and other illnesses. The collected data show that each patient has a history of being diagnosed more than one time. In addition, each time of remedy shows that each NCDs patient has more than one NCDs illness.

Contribution: This research aims to classify NCD disease in patients who are diagnosed with the multi-morbidity illness. This is one of the challenging issues in multi-label classification (MLC) research. Laboratory data obtained from patients' screening tests of NCDs diabetes, hypertension, cardiovascular, and stroke were collected. This clinical information is an important factor that indicates the diagnostic results. After that, we used the multi-label learning process to learning and predict

multiple NCDs disease at the same time. Four methods are implemented. This includes Binary relevance (BR), Classifier Chains (CC), The random k-labelsets (RAkEL), and Multi-Label k-Nearest Neighbor (ML-*K*NN). Then, we compare the performance of the methods.

The results provided predictive algorithms with the highest level of Accuracy for the multiple NCD patients. In addition, this can be used in the disease screening process, which is usually performed by doctors, for predicting NCD patients. As a result, it can improve the diagnosis process and lead to effective and comprehensive healthcare systems.

This chapter is organized as follows: Section 2.2 provides related works. Section 2.3 shows the research methodology. The Experimental results are shown in Section 2.4, and the conclusion is given in the last section.

2.2 Related work

Multi-label learning has been presented by Boutell et al. (2004). Then, Tsoumakas & Katakis. (2007a) compiled and summarized the solution of multi-label into two types, i.e., the first, adaptation method and second problem transformation methods. The researcher compared the efficiency of each sub-level method with three datasets in general to observe the effectiveness of each method. Madjarov et al. (2012a) presented an expanded experiment for the purpose of comparing multi-label methods. Three sets of methods were divided into different algorithm types, which are adaptation methods, problem transformation methods, and ensemble methods. In addition, the comparison experiments were conducted with twelve methods. The eleven datasets were tested, and each method has also been evaluated.

Runzhi Li et al. (2016) presented A Multi-Label Problem Transformation Joint Classification (MLPTJC) by solving the classification label problem on three disease data, including diabetes, hypertension, and fatty liver. This method worked in two steps. The first step was a combination of the data records. The multi multimorbidity patients were always found in multiple records in the data. The label of those patients' records was combined and represented in a single record. The second step was to classify label disease into subordinate sets to solve the imbalance problem. That technique resulted in promising accuracy of classification. In addition, there is a

similar work that conducted MLC by using algorithms, including Support Vector Machine (SVM) and Random Forest (RF), then tested with 110,300 patients' datasets. The results obtained from the experiment demonstrated that these techniques provided high accuracy. R. Li et al. (2017) a novel Ensemble Label Power-set Pruned datasets Joint Decomposition (ELPPJD) method, was presented. The advantages of this method were size balanced (SB) and label similarity (LS). The experiment was tested with three algorithms and then compared the efficiency of classification of ELPPJD with RAkEL and HOMER. The result showed that the ELPPJD method outperformed other methods.

2.3 Methodology

We used NCD patients' datasets. The four NCDs, including diabetes, hypertension, cardiovascular, and stroke, are processed in MLC by using four methods, i.e., Binary relevance (BR) (Tsoumakas & Katakis, 2007a), Classifier Chains (CC) (Read et al., 2011b), The random k-labelsets (RAkEL) (Tsoumakas & Vlahavas, 2007a) and Multi-Label K-Nearest Neighbor (ML-*K*NN) (M. L. Zhang & Zhou, 2007). The BR, CC, and RAkEL are defined to use a basic classification algorithm, which is Random Forest (Breiman, 2001). The ML-*K*NN used k-Nearest neighbor (D. Cheng et al., 2014) algorithm in the classification process shown in Figure 2. The methodology is carried out as follows: First, datasets collecting process, Second, data preparation process, Third, performance measure process, and Finally, experimental development. Each process can be explained in the subsection below.

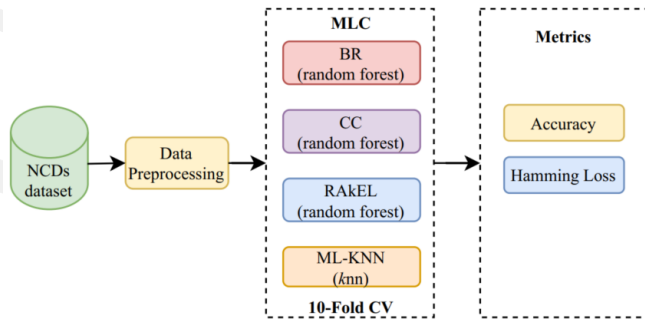


Figure 2 The multi-label classification methods used in this research.

2.3.1 Multi-label classification methods

BR (Tsoumakas & Katakis, 2007a) is a problem transformation method. Data in multi-label is made as a single label, and then it is classified by binary using the classification method in each pair. In this process, machine learning algorithms are used for classification, such as decision trees, SVM, and KNN. Classifications are processed with features (f_1, f_2, f_3, f_4, f_5,..$f1_3$) and label (l_1, l_2, l_3, ...l_4) pair by pair. Then, the next process is a probability score value is arranged from classification in each pair.

CC (Read et al., 2011b) applied in this work is similar to BR in that the data is classified in pairs amidst f_1 to l_1, but the CC has some differences. In each classification step of f_{1l1}, the f_{1l1} are connected with l_2,..l_n continuously. Then, the probability of the answer is ranked.

RAkEL (Madjarov et al., 2012a; Tsoumakas & Vlahavas, 2007a) is an ensemble method for multi-label classification. It draws m random subsets of labels with size *k* from all labels *L* and trains a label power-set classifier using each set of labels.

ML-*K*NN (Madjarov et al., 2012a; M. L. Zhang & Zhou, 2007) is an extension of the famous k-nearest neighbor (*K*NN) algorithm. First, for each test example, its *K*NN in the training set is identified. Then, according to statistical information gained from the label sets of these neighboring examples, i.e., the number of neighboring examples belonging to each possible label, the maximum a posteriori principle is used to determine the label set for the test example.

2.3.2 Dataset

We collected datasets from electronic health records at Suddhavej Hospital Faculty of Medicine, Mahasarakham University. It collected forty-three files belonging to the structure of medical and health data standards (Ministry of Public Health, 2017). Data for this experiment was 19,554 medical examinations collected from 2014 until 2019, a record representing each patient of a clinical examination. The Suddhavej Hospital has approved the collected data, and those data can not be identified back to any patients.

In this research, we are interested in the screening data for chronic disease. There are three related tables in the dataset, consisting of (i) the SERVICE

collected from historical services of the patients, (ii) NCDSCREEN, a table that collects chronic disease screening, and (iii) Diagnosis_OPD collected from the diagnostic results of a patient. All the data tables are related and shown in Figure 3.

Figure 3 Database relationship of medical and health data standards.

This work is only interested in the results of the data analysis diagnosed by the ICD-10 standard (WHO, 2016) with 4 NCDs, including diabetes, hypertension, cardiovascular and stroke. The collected data is prepared to process multi-label, as shown in Table 1.

Table 1 Multi-Label Physical Examination Records Description.

Records	Diabetes	Hypertension	Cardiovascular	Stroke
R_1	*	*		*
R_2		*	*	
R_3	*	*		*
…				
R_n	*		*	*

2.3.3 Data Preprocessing

In this work, the datasets are composed of two tables, according to NCDSCREEN and Diagnosis_OPD. The NCDSCREEN records the information from the patients, comprising twenty-two columns. In addition, we eliminate the data that is not related to our work in this process, only thirteen fields' attributes are used. Those attributes indicate the diagnosis of NCDs. Diagnosis_OPD is a table collected from the diagnostic results of medical services patients. There are ten columns in the table. In this step, the data is prepared as follows.

Data Cleaning is firstly performed. A group of data columns is eliminated. This group of columns is considered not to be explicit factors for the classification, such as the hospital code, date of examination, type of service date, month, and year of data update. For the missing data, we use the correction to replace the missing value by inserting the median of all data in that attribute (Ezzine & Benhlima, 2018).

Next, data Integration is performed. This is to combine two table schemas. We select only the DIAGTYPE attributes as the diagnostic types and the DIAGCODE referencing the ICD-10-TM disease code. Then, we filter the data to select only four NCDs, i.e., diabetes, hypertension, cardiovascular, and stroke. An example of the selected data is shown in Table 2. The attributes are classified into 13 groups, and data types are demonstrated in Table 3. The final summary of the dataset used in the experiment is shown in Table 4.

Finally, data Transformation is carried out by converting data to the specified data range. We apply the Min-Max Normalization method to normalize the data in the range of 0-1.

Table 2 Equation 1 ICD-10 Code for the four NCDs.

Disease	Diagnosis code (ICD-10)
Diabetes	E10, E11, E12, E14
Hypertension	I10, I11, I12, I13, I14, I15
Cardiovascular	I20, I21, I22, I23, I24, I25
Stroke	I60, I61, I62, I63, I64

Table 3 The Detail Information of NCDs Dataset.

Attributes	Description	Data Type
SMOKE	Smoking history	Nominal
ALCOHOL	Alcoholic drinking history	Nominal
DMFAMILY	Diabetes history in direct relatives	Nominal
HTFAMILY	Hypertension history in direct relatives	Nominal
WEIGHT	Weight	Numeric
HEIGHT	Height	Numeric
WAIST_CM	Waist circumference	Numeric
SBP_1	Systolic Blood Pressure: SBP 1st test	Numeric
DBP_1	Diastolic Blood Pressure: DBP 1st test	Numeric
SBP_2	Systolic Blood Pressure: SBP 2nd test	Numeric
DBP_2	Diastolic Blood Pressure: DBP 2nd test	Numeric
BSLEVEL	Blood sugar levels	Numeric
BSTEST	Methods of checking blood sugar	Nominal
Label_Diabetes	Diabetes diagnosis (0=negative, 1=positive)	Nominal
Label_Hypertension	Hypertension diagnosis (0= negative, 1= positive)	Nominal

Table 3. The Detail Information of NCDs Dataset (Cont.).

Attributes	Description	Data Type
Label_Cardiovascular	Cardiovascular diagnosis (0= negative, 1= positive)	Nominal
Label_Stroke	Stroke diagnosis (0= negative, 1= positive)	Nominal

Table 4 Summarize the NCDs Dataset used in the Experiment.

Instances	Features	Label	Label set	Card	Dens
19,554	13	4	15	0.151	0.038

In general, the input to the multi-label algorithms is a dataset S, with N instances $T_i, i = 1,\ldots,N$, chosen from a domain X with fixed, arbitrary and unknown distribution D, of the form (x_i, Y_i), with $i = 1,\ldots,N$, for some unknown function $f(x) = Y$. L is the set of possible labels of the domain D, and $Y_i \subseteq L, i.e., Y_i$ is the set of labels of the ith instance (Bernardini et al., 2014).

The number of labels $|L|$ is frequently seen as a parameter that influences the performance of different multi-label methods. There are two measures for evaluating the characteristics of a dataset, objects of this study: cardinality ($Card$) and density ($Dens$), defined as:

$$Card = \frac{1}{N}\sum_{i=1}^{N} |Y_i| \quad (1)$$

$$Dens = \frac{1}{N}\sum_{i=1}^{N} \frac{|Y_i|}{|L|} \quad (2)$$

2.3.4 Evaluation Measures

Measuring the effectiveness of MLC methods can be determined by normal single-label classification. In the multilabel field, Accuracy is defined as Equation. 3. the proportion between the number of correctly predicted labels and the total number of active labels in both true and the predicted label sets. The equation below applied to each instance and averaged (Herrera, Charte, Rivera, del Jesus, et al., 2016).

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (3)$$

The Hamming loss as Equation. 4. measures the average error for all predicted labels, the symmetric difference between sets *Y* and *Z*, and is equivalent to the Exclusive OR (XOR) logic operation. Hamming loss values indicate a better classification performance (Tanaka et al., 2015)

$$Hamming\ loss = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n} |Y_i \,\Delta\, Z_i| \quad (4)$$

n denotes the number of data instances, *k* represents the total number of elements, Y_i is a subset of predicted labels for each instance, and Z_i is the actual subset of labels (Herrera, Charte, Rivera, del Jesus, et al., 2016).

2.4 Experimental

2.4.1 Experimental Setup

We used datasets in these experiments to compose thirteen features and four labels using four methods of multi-label classification, including BR, CC, RAkEL and ML-*K*NN. The BR, CC, and RAkEL were defined to use a basic classification algorithm, which is Random Forest. ML-*K*NN used the k-Nearest neighbor algorithm in the classification process. The MLC tool is Meka software (Read et al., 2016), which is an extension of the Weka program used in BR, CC, RAkEL methods. In addition, the ML-*K*NN method used the MULAN framework (Tsoumakas, Spyromitros-Xioufis, et al., 2011) for the implementation of experimental results. This work used an evaluation model with 10-fold cross-validation.

The following algorithm determines the parameters: BR and CC do not provide the parameter values. RAkEL algorithm, the number of labels in each subset (k) is 3, the number of subsets (m) is 10, and the number of frequent label sets to subsample from infrequent label sets (n) is 0. ML-*K*NN algorithm, the number of neighbors (k) is 3.

The parameters in a Random Forest classification algorithm are initialized as follows: the number of trees (numTrees) is 1000, of bag score is set as false, the maximum depth of the trees (maxDepth) is 0 for unlimited depth, the number of attributes used in random selection (numFeatures) is 0.

2.4.2 Experimental Result

The experiments were set up using the four MLC methods to learn the given datasets. The efficiency of each method was measured by accuracy and hamming loss, which is shown in Table 5. The explanations are as follows. First, the BR method results in an 84.37% accuracy rate, and the hamming loss value is 0.0504.

Second, the CC method provides an 89.96% accuracy rate; hamming loss value is 0.0377. Third, the RAkEL method outputs 91.07%, and of accuracy rate, hamming loss value is 0.0377. Fourth, the ML-*K*NN method has an accuracy rate of 72.10%; hamming loss is 0.0874.

Table 5 Performance evaluation for different multi-label methods.

Methods	**Accuracy (%) ± S.D.**	**Hamming Loss ± S.D.**
BR	84.37 ± 0.0065	0.0504 ± 0.0024
CC	89.96 ± 0.0061	**0.0377** ± 0.0022
RAkEL	**91.07** ± 0.0074	**0.0377** ± 0.0025
ML-*K*NN	72.11 ± 0.0084	0.0874 ± 0.0030

We compared each MLC method, as can demonstrate in Figure 4. The result shows that the RAkEL method gives the highest efficiency, considered by accuracy value, 91.07% of accuracy. For the hamming loss method, as shown in Figure 5, both RAkEL and CC obtained the result of marginal loss closing to 0, which is 0.0377. Both methods show the lowest prediction errors than other methods.

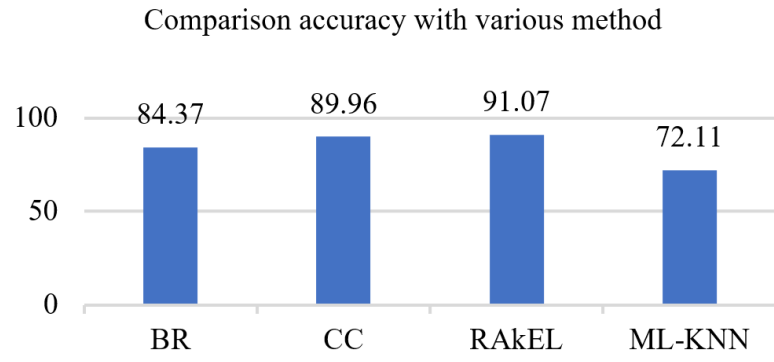


Figure 4 Performance accuracy comparison with multi-label classification methods.

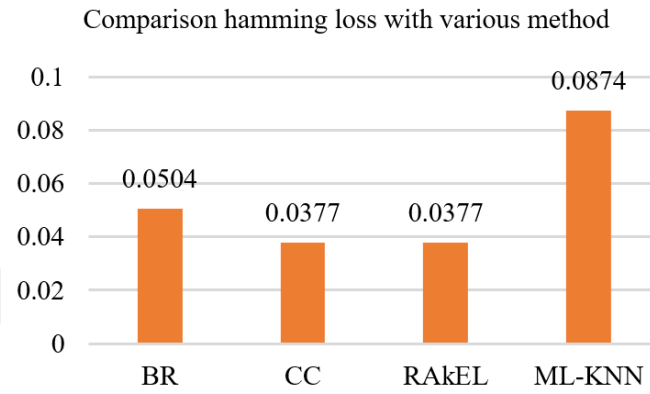


Figure 5 Performance hamming loss comparison with multi-label classification methods.

2.5 Conclusion

This chapter investigates the traditional comparative methods of multi-label classification. We used four MLC methods BR, CC, RAkEL, and ML-*K*NN. Used the NCDs dataset from Suthavej Hospital to generate the classification models and construct the evaluation experiments. The results obtained from the experiments were compared to investigate the empirical significance of the performance of different classification methods. The result demonstrates the efficiency of each method. The RAkEL method is the most effective method, with an accuracy rate of 91.07%, the highest rate compared with the other three methods.

The next chapter proposed a method to improve the MLC performance using the feature reconstruction method. The work applies an AutoEncoder network to capture the relationship between data features and their labels. The experimental results demonstrate that the proposed technique provided better performance for the classification algorithm.

Chapter 3

Improving Multi-label Classification using Feature Reconstruction Methods

3.1 Introduction

Multi-label classification (MLC) is a supervised classification method that essentially takes input instances and classifies them to a set of target values (labels) simultaneously (Chandran & Panicker, 2017; Prajapati & Thakkar, 2021). In general, the search space of the MLC problem is large compared with that of multi-class classification (MCC) and exponentially when the number of possible labels increases (Bogatinovski et al., 2021). In addition, MLC is a non-mutual exclusive classifier. Therefore, MLC can produce complex decision boundaries. In addition, the number of data instances used in training processes can affect the performance of the classification. Inadequate data instances, compared to the number of class labels, can produce poor classification results (Alazaidah & Ahmad, 2016). MLC problems can be solved by transforming the problems into a set of single multi-class classifications. This transformation approach has been applied and applicable to various MLC problems, which is known as the Problem Transformation Method (PTM) (Alluwaici et al., 2020; Pushpa & Karpagavalli, 2017). MLC is converted to the n-class problem, where n is the number of the class label extracted from the set of the multi-class label. In addition to PTM, the Adaptive Method (AM) applying the available classification technique (for multi-class problems) has also been implemented to solve various MLC problems.

In the past decades, multi-label learning has gained more attention in the research into solving MLC problems (Boutell et al., 2004). Initially, Tsoumakas & Katakis. (2007b) compiled and summarized the solutions of MLC into two categories, i.e., (i) adaptation method and (ii) problem transformation methods. Madjarov et al. (2012b) presented an expanded experiment to compare the performance of different types of classification algorithms for MLC (Sangkatip & Phuboon-Ob, 2020). The

study derived and experimented with the three classification-based algorithm groups: PTM, EM, and Ensemble Methods (EM). In the PTM, Binary Relevance (BR) method and Label Powerset (LP) method were implemented, which transforms the MLC problem into basis problem subsets of binary-classification problems. Then, the aggregation strategy was applied to obtain the final label set. In the AM, Decision Tree algorithms, for example, have been applied to carry out the classification of multi-label data (Sousa & Gama, 2016). The C4.5 algorithm was one of the most commonly used algorithms deployed and is known as ML-C4.5 (Moral García et al., 2019). The K-Nearest Neighbors algorithm was also applied to MLC problems. The technique considers a set of neighbor data instances to derive the actual label set of a given data instance. This technique is known as ML-*K*NN (M. L. Zhang & Zhou, 2007). Apart from that, Neural Network-based methods are also reported in the literature that has been used effectively to compile the MLC problem (M. L. Zhang & Zhou, 2006). In the EM, MLC was decomposed into smaller problems. Then, each sub-problem was handled separately before they were ensembled to gain the final classification results using, for instance, voting schemes (Jin et al., 2017; Read et al., 2008; Tsoumakas & Vlahavas, 2007a).

Several past studies have attempted to improve the efficiency of MLC by reducing the size of the data instances. Zhang. (2011) proposed that LIFT used *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 established by creating additional attributes of the data (Gao et al., 2020). Huang et al. (2018) proposed a technique to learn the dispersion of label attributes, including common attributes. They applied double-label correlation to differentiate labels for each category. Multi-label classifiers are built on low-dimensional visualizations with these learned attributes. From that perspective, recent research into MLC has gained more attention in developing a future engineering method to improve the data features, assisting in the classification processes (Guozhu & Huan, 2018; Hafeez et al., 2021). Feature engineering can be divided into several categories, for example, feature transformation, feature generation, feature selection, and feature reconstruction

(Guozhu & Huan, 2018). Deep learning approaches are also active in recent years. Feature extraction and generation are usually some of the applicable techniques that have been implemented to improve the quality of data features. Using Convolutional Neural Networks (CNNs) for feature extraction and generation, CNNs map the input data space to another data representation based-on training data instances (Emmert-Streib et al., 2020). Dimensional reduction is one of the techniques used to transform data features. There are two categories of dimensionality reduction methods. One is Feature Selection (FS) and feature transformation (FT). Feature selection keeps only useful features and dismisses others, while feature transformation constructs a new but smaller number of features out of the original ones (Deng et al., 2013b). The current FT method can be applied by implementing, for example, deep learning algorithms (Patterson & Gibson, 2017), and unsupervised network algorithms, which learn to encode data to extract the relationships of the data. Y. Cheng et al. (2019) used a deep learning technique to build and extract relationships between attributes and labels in a multi-label classification. Feature reconstruction, a transformation process, can be considered a tool to generate a set of new feature sets (based on the original data features). The reconstructed features are anticipated to be compact and descriptive, which can be used in the classification process. This work applies the AutoEncoder approach to learn insight into the data features and construct more meaningful active features.

The work is organized as follows: Section 3.2 explains MLC and the generally applicable techniques for solving MLC. In Section 3.3, the proposed approach is described. The datasets used in this work are also delineated and explored. The feature reconstruction method is subsequently explained. Section 3.4 demonstrates that experiments were conducted to evaluate the performance of the proposed feature reconstruction method. Section 3.5 describes the result and discussion before the conclusion of this work given in Section 3.6.

3.2 Multi-label Classification

The task of MLC can be viewed as an instantiation of the structure output prediction paradigm. The goal is to define a set of labels for each data instance. Let X be a space of data instances comprising n data instances $\boldsymbol{x}$, i.e. $\forall \boldsymbol{x} \in X, \boldsymbol{x} = x_1, \dots x_d$ (where d is the number of instance features) a set of d-dimensional features divided from x , and a set p a possible label space $Y = \boldsymbol{y_1}, \dots \boldsymbol{y_p}$, i.e. $\boldsymbol{y} = y_1, .. y_m$ where $y = 0,1$ and m denotes the dimension of the labels $\boldsymbol{y}$ associated with **x**, as demonstrated in Table 6.

Table 6 Representation of data instance.

$\boldsymbol{X}$					$\boldsymbol{Y}$
$\boldsymbol{x}_1$	x_{11}	x_{12}	$\cdots$	x_{1d}	$\boldsymbol{y}_1 = y_{11}, \dots, y_{1m}$
$\boldsymbol{x}_2$	x_{21}	x_{22}	$\cdots$	x_{2d}	$\boldsymbol{y}_2 = y_{21}, \dots, y_{2m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\cdots$	$\vdots$
$\boldsymbol{x}_n$	x_{n1}	x_{n2}	$\vdots$	x_{nd}	$\boldsymbol{y}_n = y_{n1}, \dots, y_{nm}$

We denote the quantity L as a loss value for learning models. Therefore, MLC is objected to finding h such that:

$$\min_{L} h: X \to 2^Y \tag{5}$$

The MLC methods are separated into two categories: problem transformation and algorithm adaptation. The group of problem transformation methods approaches the problem of MLC by transforming the multi-label dataset into one or multiple datasets. These datasets are then approached with simpler, single-target machine learning methods and built into one or multiple single target models. At prediction time, it is required that all built models are invoked to generate the prediction for the test example. Algorithm adaptation methods include some adaptation of the training and prediction phases of the single target methods toward handling multiple labels simultaneously. For example, trees change the heuristic used when creating the splits, Support Vector Machines (SVMs) employ additional threshold techniques, etc. The adaptations provide a mechanism to handle the dependency between the labels directly. Their grouping is based on the underlying paradigm being adapted. The

literature recognizes five defined groups of algorithm adaptation methods according to the performed adaptation: trees, neural networks, support vector machines, instance-based and probabilistic. There are additional methods that utilize various approaches from other domains, e.g., genetic programming, but they lack a common ground to unite them and are characterized as an unspecified group of methods.

3.2.1 Transformation-Based Classifiers

A Transformation-Based Classifier (TBC) transforms an MLC into a simpler classification problem, which can be potentially solved by single-label multi-class classification. The classification essentially provides possible values for the transformed class label, the set of distinct unique subsets of the label in the original data instance (Read et al., 2014). A number of techniques have been proposed. Label Powerset generally generates a new set of single class labels. Given a data instance x with a corresponding label $\boldsymbol{y} = 1,0,0,1,1$ in the original MLC problem shown in Table 7, the Label Powerset will generally transform the data instance into a new label $y_{1,4,5}$ which can deliberately be used with available multi-class classifier techniques. An example of the problem transformation results is shown in Table 8.

Table 7 Example of label powerset multi-label transformation.

X	Y
$\boldsymbol{x}_1$	$\boldsymbol{y}_1$= {1,0,0,1,1}
$\boldsymbol{x}_2$	$\boldsymbol{y}_2$ = {1,0,0,1,1}
$\boldsymbol{x}_3$	$\boldsymbol{y}_3$ = {1,0,0,0,0}
$\boldsymbol{x}_4$	$\boldsymbol{y}_4$ = {1,1,0,0,0}
$\boldsymbol{x}_5$	$\boldsymbol{y}_5$ = {1,0,0,0,0}

Table 8 The transformed problem becomes a 3-class problem.

X	Y
$\boldsymbol{x}_1$	$\boldsymbol{y}_1 = \{y_{1,4,5}\}$
$\boldsymbol{x}_2$	$\boldsymbol{y}_2 = \{y_{1,4,5}\}$
$\boldsymbol{x}_3$	$\boldsymbol{y}_3 = \{y_1\}$
$\boldsymbol{x}_4$	$\boldsymbol{y}_4 = \{y_{1,2}\}$
$\boldsymbol{x}_5$	$\boldsymbol{y}_5 = \{y_1\}$

In addition to Label Powerset, Binary Relevance (BR) method is one of the TBC methods that are commonly used to solve problems. BR decomposes an MLC problem into distinct single-label binary classification problems, one for each of the m labels in the set $\boldsymbol{y} = y_1, \dots, y_m$ (Cherman et al., 2011). In the learning process, the original multi-label training dataset is transformed to m datasets, and each of them is associated with a binary class-label obtained from the original y. After the multi-label data has been transformed, a set of q binary classifiers $H_j(x), j = 1..m$ is constructed using the new m training dataset. The BR generates a set of m classifiers as follows:

$$H = M_{y_j}(\boldsymbol{x}, \boldsymbol{y}_j) \rightarrow y' \in 0,1 | y_j \in y, j = 1, \dots, m \quad (6)$$

3.2.2 Adaptation-Based Classifiers

A lazy classification is one of the available techniques that has been applied to solve MLC. Multi-Label K-Nearest Neighbor (ML-*K*NN) is introduced for the purpose (M. L. Zhang & Zhou, 2007). The ML-*K*NN determines a label set of an instance (x) of the unknown label ($y(x) \subseteq y$) by utilizing the Maximum A Posteriori (MAP) method to predict the label set of x. Given an unknown label set x, ML-*K*NN examines k neighbors of x (based-on a distance metric) and count a number of neighbor (s) belonging to each class y_i.

$$P(y_i|s) = \frac{P(s|y_i)P(y_i)}{P(s)}. \quad (7)$$

For each label y_i, KL-*K*NN generates a h_i classifier to predict the final label set:

$$h_i = \begin{cases} 1 & P(y_i = 1|s) > P(y_i = 0|s) \\ 0 & otherwise. \end{cases} \quad (8)$$

Support Vector Machine (SVM) has also been applied in Adaptation-Based Classifiers (ABC) to solve MLC (Cortes & Vapnik, 1995). Conventionally, SVM is introduced to cope with binary classification problems. However, in multi-label classification, a ranking SVM was proposed RANK-SVM method (Elisseeff & Weston, 2001).

3.2.3 Ensemble-Based Classifiers

Ensemble-based Classifier (EBC) transforms an MLC problem into a set of smaller problems p (ensemble on a subset of the problems (Gibaja et al., 2016; Rokach et al., 2013; Tsoumakas & Katakis, 2007b). Each of the problems is solved separately as a subset classifier. The results of the subset classifiers are aggregated (assembled) to produce a final decision on the classification. Random k-Labelsets (RAkEL) is one of EBCs (Kimura et al., 2016; Tsoumakas & Katakis, 2007b). The technique builds a random subset of the original labels to learn a single-label classifier (binary) for the prediction of each element in the powerset of the subset. To illustrate the basis of the basic idea of the RAkEL, consider Table 9, which shows four random subsets ($M_j, j = 1, \dots, 2^k$ and k is a size of feature subset obtained from y) of the MLC problems for $k = 2$. For each subset problem, k binary classification is performed. Then, the final decision is aggregated by a voting mechanism.

Table 9 Example of problems subsets in RAkEL technique.

Model	y_1	y_2	y_3	y_4	y_5
M_1 (1,5)	1	-	-	-	1
M_2 (2,4)	-	0	-	1	-
M_3 (2,3)	-	1	1	-	-
M_4 (4,5)	-	-	-	1	0
avg-votes	1/1	1/2	1/1	2/2	1/2
prediction	1	0	1	1	0

Table. 9 Example of problems subsets in RAkEL technique (k=2) applying on data presented in Table.7 The final decision is made by thresholding (using a predetermined, e.g., $\tau = 0.5$) the average of votes (AV_i) of each label dimension (y_i). The prediction for $y_i$is 1 when $AV_i > \tau$ and 0 otherwise.

3.3 Methodology

Solving MLC is essentially a challenging task. Many methods usually undergo a design of the algorithms that cope with the problem and yet produce promising classification results (Gao et al., 2020; Huang et al., 2018; M.-L. Zhang, 2011). This work focuses on the feature engineer-based method, where the features of data instances are explored and transformed to a compact form used in a subsequent classification process. Therefore, in this section, we provide enough details of the datasets used in this study and the proposed method.

3.3.1 Dataset

The data used in this work was collected from the Mulan datasets website (Tsoumakas, Spyromitros-Xioufis, et al., 2011). There are eight standard datasets comprising different data domains, as demonstrated in Table 10.

The number of feature dimensions (d) in the dataset is varied. In addition, each dataset is associated with a different number of class labels. For example, the yeast dataset has 2417 data instances (the biggest dataset) with 103 feature dimensions and 14 subset features. The cardinality of the dataset denotes

variation of each class-labels in the dataset. The density of the dataset explains the variation of the class labels with respect to the number of labels in the dataset.

Table 10 Multi-label Datasets.

Datasets	Domain	Instances	Features	Labels	Cardinality	Density
birds	audio	645	260	19	1.014	0.053
enron	text	1702	1001	53	3.378	0.064
emotions	music	593	72	6	1.869	0.311
medical	text	978	1449	45	1.245	0.028
yeast	biology	2417	103	14	4.237	0.303
scene	image	2407	294	6	1.074	0.179
cal500	music	502	68	174	26.044	0.15
foodtruck	recommend	407	21	12	2.29	0.191

In general, the Cardinality is the average number of labels per example, defined in Equation 1.

The Density is the number of labels per sample divided by the total number of labels, defined in Equation 2.

3.3.2 Feature Reconstruction using AutoEncoder

The work proposes a technique that transforms a set of Feature Transform (FT) of given data instances into a compacted feature space (X'). To achieve this task, we introduce a transformation function $t: \boldsymbol{x} \rightarrow \boldsymbol{x'}$, $\boldsymbol{x'} = x'_1, \ldots, x'_k$ where k is a dimension of the transformed features and $k \ll d$. We adopt auto-encoder techniques as the transformation function (t) that encodes the input data features. The proposed technique compresses the input data instances with an encoder module using an Auto-Encoder technique (EN) (Liou et al., 2014). In addition, we introduce an extension mechanism that in-cooperates the Target-label to the auto-ENcoder network (TEN) during the network training process. The extension process can potentially compact the original features and maintain the context of the transformed features with respect to the original data labels. There are two processes that are carried out to classify the data instances using the proposed feature

reconstruction method, i.e., feature reconstruction and classification, illustrated in Figure 6.

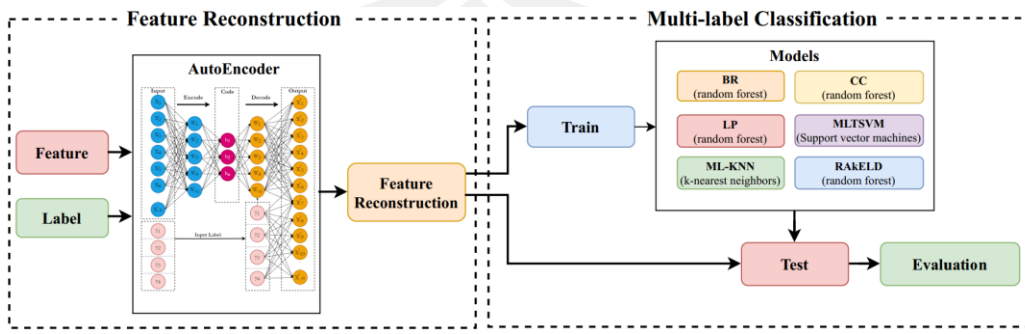


Figure 6 The overall process of the feature reconstructing for solving MLC.

3.3.2.1 Feature Reconstruction

AutoEncoder technique is applied in this work to encode the input data instances as the main procedure for compacting the original features of the data ($\boldsymbol{x}$). The AutoEncoder is a Neural Network (NN) that can be used to learn and derive the representation of data. The network is decomposed by two main modules, i.e., the encoder and decoder module, as depicted in Figure 7.

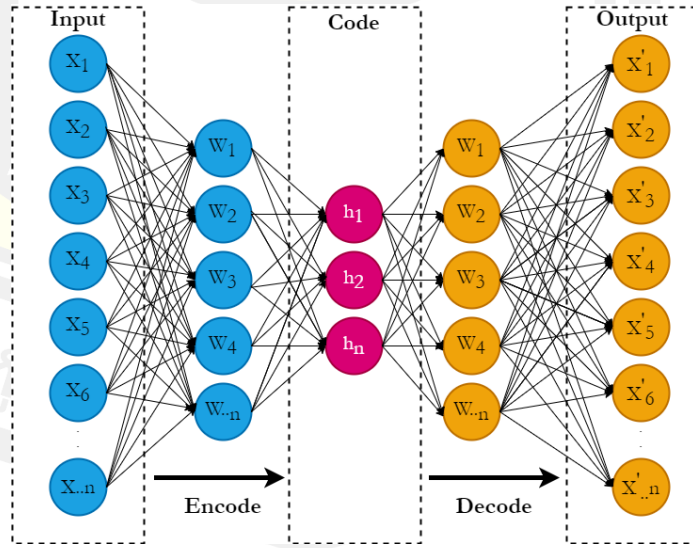


Figure 7 AutoEncoder architecture EN.

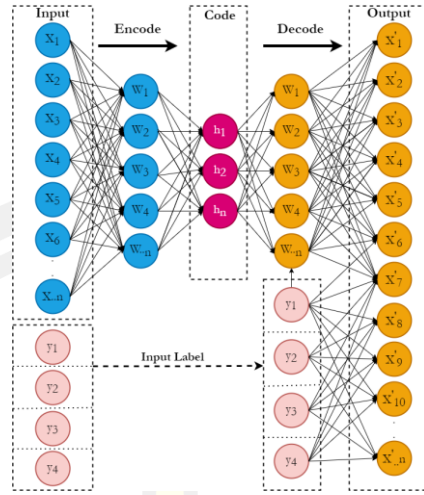


Figure 8 AutoEncoder architecture TEN.

The encoder module encodes the input data instance while the decoder attempts to decode the encoded data. The decoder actively tries to decode the encoded data to be identical to the original data representation during the training process. This process can be performed through an optimization approach, aiming to minimize a certain criterion. In this work, we denote L as a loss that measures the difference between the input instance (x) and the decoded data (x'). To obtain both solid encoder and decoder module, we define a loss function as follows:

$$\min_{W,W',b,b'} L(\boldsymbol{x}, \boldsymbol{x}') = ||x - \sigma'(W'(\sigma(W\boldsymbol{x} + b)) + b')||^2 \quad (9)$$

This loss function is minimized with respect the network parameters (W, W', b, b') and σ denotes activation functions (Kimura et al., 2016). After the training process, the encoder module will be used to reconstruct compact features for the subsequent classification procedure.

The constructed features using the encoder module from the trained decoder cannot potentially be applicable to represent the data features. Therefore, in this work, we integrate the class labels (**y**) as a set of the augmented node to the input data instances, and we define it as the TEN methods. Then, the output layer (associated with the decoder module) is compiled to generate $|x| + |y|$

output nodes, as illustrated in Figure 8. The optimization can subsequently be performed using the loss function below:

$$\min_{W,W',b,b'} L(\boldsymbol{x},\boldsymbol{x}') = ||\boldsymbol{x}\frown\boldsymbol{y} - \sigma'(W'(\sigma(W\boldsymbol{x}+b))+b')||^2 \quad (10)$$

$\boldsymbol{x}\frown\boldsymbol{y}$ denotes the concatenated vectors between an instance $\boldsymbol{x}$ and its associated label $\boldsymbol{y}$. To generate a discriminative feature from the network (W, b) , we reconstruct the features using a reconstruction, $\tau(.)$, from the input original feature (x) as follow:

$$\tau(\boldsymbol{x}) = \sigma(W\boldsymbol{x}+b). \quad (11)$$

3.3.2.1 Multi-label Classification

The previous section provides the details of the feature reconstruction used in this research work. The reconstructed features $(\tau(\boldsymbol{x}))$ is then fed to the classification process $f:\tau(\boldsymbol{x}) \rightarrow \boldsymbol{y}'$, where $f(.)$ is a mapping function or a classifier. This work uses various classification techniques to classify the original data $\boldsymbol{x}$ and the reconstated $\tau(\boldsymbol{x})$. The details of the classification settings and experiments will be explained in the next section.

3.4 Experiment Setup and Evaluation Metrics

The previous section explained the proposed method for constructing a new feature subset. The input data instances are fed to the encoder module (EN and TEN) to generate compact features. Then, the classification is carried out. This section provides the details of the experiment conducted to evaluate the performance of the proposed method.

To evaluate the performance of the proposed feature construct method, we used six MLC methods to classify datasets through instance transformations. These classification techniques were used to examine the effectiveness of the proposed method when experimenting with various common MLC classification techniques, i.e., PTM, Adaptation method, and Ensemble technique. Binary relevance (BR) and classifier Chains (CC) (Read et al., 2011a) were used in the experiments. These two classification techniques are based on the problem transformation method). In

addition to the PTM, the Label Powerset (LP) was also implemented in this work as the technique is the fundamental method used for MLC problems. The Adaptation method, i.e., MLTSVM (W.-J. Chen et al., 2016) and ML-*K*NN, were also utilized. Finally, The RakELd (Tsoumakas, Katakis, et al., 2011), an Ensemble-based technique for MLC, was used in the experiments.

For each dataset, the dataset was divided into train and test sets. The train data was used to train the AutoEncoder. We separated 60% of the data instance from the dataset to construct the training process. The remaining 40% of the data instances were used to test the performance of classification performance (by all six classifiers.)

In the experiment, we utilized Scikit-multilearn as a primary tool to conduct various experiments (Szymański & Kajdanowicz, 2017). We choose ten common evaluation metrics for MLC (Wu & Zhou, 2017). These evaluation metrics cover both example-based metrics and label-based metrics, namely, Precision, Recall, F1, Macro Precision (Macro P), Macro Recall (Macro R), Macro F1, Micro Precision (Micro P), Micro Recall (Micro R), Micro F1, and Hamming Loss (H Loss). To the shake of the representation simplicity, these evaluation metrics are denoted as Precision, Recall, F1, Macro P, Macro R, Macro F1, Micro P, Micro R, Micro F1, and H Loss for convenience. Precision and Recall are defined as the average proportion between the number of correctly predicted labels. The measurement metrics are defined as follows equation 12-21.

For each classifier, true positives (tp_j), true negatives (tn_j), false positives (fp_j), and false negatives (fn_j) obtained (based on the metrics) are calculated for each label $y: j = 1 \ldots m,$. Macro F_1 is essentially the harmonic mean obtained from Precision and Recall based on an average of each label y_j and an average of over all labels. In addition, Micro F_1 is the harmonic mean of Micro derived from Precision and Micro Recall in the above definition.

$$\text{Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{12}$$

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{13}$$

$$\text{F1} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \tag{14}$$

$$\text{Hamming Loss} = \frac{1}{|N| \cdot |L|}\sum_{i=1}^{|N|}\sum_{j=1}^{|L|} xor(y_{ij}, \hat{y}_{ij}) \tag{15}$$

$$\text{Micro Precision } (MiP) = \frac{\sum_{j=1}^{m} tp_j}{\sum_{j=1}^{m} tp_j + \sum_{j=1}^{m} fp_j} \tag{16}$$

$$\text{Micro Recall (MiR)} = \frac{\sum_{j=1}^{m} tp_j}{\sum_{j=1}^{m} tp_j + \sum_{j=1}^{m} fn_j} \tag{17}$$

$$\text{Micro F1} = \frac{2 \times MiR \times MiP}{MiR + Mip} \tag{18}$$

$$\text{Macro Precision} = \frac{1}{m}\sum_{j=1}^{m}\frac{tp_j}{tp_j + fp_j} \tag{19}$$

$$\text{Macro Recall} = \frac{1}{m}\sum_{j=1}^{m}\frac{tp_j}{tp_j + fn_j} \tag{20}$$

$$\text{Macro F1} = \frac{1}{m}\sum_{j=1}^{m}\frac{2 \times R_j \times P_j}{R_j + P_j} \tag{21}$$

3.5 Results and Discussion

After training the AutoEncoder, we conducted two separate experiments. The first experiment aimed to examine the efficiency of the feature construction of the two techniques, i.e., EN and TEN. In addition, we experimented on each dataset separately. The experimental results are listed in the following tables (↑ indicates the higher value, the better, and ↓ the lower, the better).

Table. 11-18 demonstrate the results from the experiments carried out on the eight different datasets using EN and TEN for the feature reconstruction. From the experimental results, we can observe that the construction method TEN outperforms EN for almost all of the datasets for all measurement metrics. TEN results better or the same outcomes for all datasets and classifiers. Consider Yeast and Emotion datasets (Table. 11 and Table. 12); TEN tends to produce better results than EN for all different classifiers and evaluation matrices. Based on the data description (shown in Table. 10), Yeast and Emotions datasets are the only two datasets with a high-density value (>0.3). The density practically measures the dispersion of the data. With the MLC dataset, the density signifies the distribution of the data labels.

High density accounts for low label dispersion, well presented. Compared to other datasets, e.g., the Birds and Medical datasets, The EN and TEN provide marginally the same results for some classifiers. Using TEN for the Yeast dataset, the best performance (measured by F1) is 78.0% obtained from the BR technique. Moreover, the best performance resulting from the Emotions dataset using TEN is at 60.0%, respectively.

Table 11 Comparative results for Yeast dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.69±0.03	**0.85±0.03**	0.66±0.02	**0.83±0.02**	0.61±0.02	**0.79±0.03**	0.63±0.02	**0.83±0.03**	0.59±0.02	**0.78±0.03**	0.66±0.02	**0.82±0.02**
Recall↑	0.52±0.01	**0.76±0.00**	0.52±0.01	**0.76±0.02**	0.56±0.02	**0.74±0.01**	0.59±0.02	**0.77±0.01**	0.51±0.01	**0.75±0.01**	0.56±0.02	**0.76±0.00**
F1↑	0.56±0.02	**0.78±0.01**	0.55±0.01	**0.77±0.02**	0.56±0.02	**0.74±0.02**	0.59±0.02	**0.78±0.02**	0.52±0.01	**0.74±0.01**	0.58±0.02	**0.77±0.01**
Macro P↑	0.42±0.03	**0.75±0.04**	0.39±0.01	**0.73±0.04**	0.40±0.02	**0.66±0.03**	0.34±0.02	**0.71±0.02**	0.37±0.01	**0.65±0.04**	0.42±0.06	**0.72±0.04**
Macro R↑	0.29±0.00	**0.54±0.01**	0.30±0.01	**0.53±0.02**	0.35±0.01	**0.54±0.01**	0.34±0.01	**0.54±0.01**	0.32±0.00	**0.59±0.02**	0.33±0.01	**0.54±0.01**
Macro F1↑	0.31±0.01	**0.59±0.01**	0.31±0.00	**0.58±0.02**	0.35±0.01	**0.57±0.01**	0.32±0.01	**0.58±0.01**	0.34±0.01	**0.61±0.02**	0.33±0.02	**0.58±0.01**
Micro P↑	0.70±0.03	**0.87±0.03**	0.67±0.03	**0.85±0.02**	0.62±0.02	**0.79±0.02**	0.63±0.03	**0.84±0.03**	0.59±0.02	**0.78±0.03**	0.66±0.02	**0.83±0.02**
Micro R↑	0.52±0.01	**0.75±0.01**	0.52±0.01	**0.74±0.02**	0.56±0.02	**0.73±0.01**	0.58±0.02	**0.75±0.01**	0.51±0.01	**0.74±0.01**	0.56±0.01	**0.75±0.01**
Micro F1↑	0.59±0.02	**0.80±0.01**	0.58±0.01	**0.79±0.01**	0.58±0.02	**0.76±0.01**	0.61±0.02	**0.79±0.01**	0.55±0.01	**0.76±0.01**	0.60±0.02	**0.79±0.01**
H Loss↓	0.21±0.01	**0.11±0.00**	0.22±0.00	**0.12±0.01**	0.24±0.01	**0.14±0.00**	0.23±0.01	**0.12±0.01**	0.25±0.01	**0.14±0.01**	0.22±0.01	**0.12±0.00**

Table 12 Comparative results for Emotions dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.36±0.07	**0.56±0.01**	0.38±0.05	**0.60±0.05**	0.40±0.05	**0.57±0.03**	0.36±0.06	**0.60±0.04**	0.32±0.05	**0.63±0.03**	0.39±0.03	**0.55±0.04**
Recall↑	0.31±0.04	**0.52±0.05**	0.35±0.06	**0.57±0.05**	0.42±0.04	**0.56±0.06**	0.47±0.08	**0.68±0.08**	0.27±0.08	**0.63±0.07**	0.38±0.08	**0.56±0.04**
F1↑	0.31±0.04	**0.51±0.02**	0.34±0.04	**0.55±0.01**	0.39±0.03	**0.54±0.03**	0.39±0.05	**0.62±0.04**	0.27±0.06	**0.60±0.04**	0.36±0.05	**0.53±0.03**
Macro P↑	0.40±0.09	**0.62±0.02**	0.36±0.08	**0.62±0.04**	0.40±0.04	**0.57±0.05**	0.15±0.03	**0.59±0.04**	0.32±0.07	**0.63±0.03**	0.39±0.06	**0.60±0.04**
Macro R↑	0.29±0.04	**0.51±0.06**	0.33±0.07	**0.58±0.07**	0.39±0.04	**0.58±0.08**	0.43±0.08	**0.70±0.07**	0.25±0.08	**0.62±0.08**	0.36±0.07	**0.57±0.05**
Macro F1↑	0.32±0.05	**0.54±0.03**	0.33±0.07	**0.57±0.03**	0.39±0.04	**0.55±0.03**	0.22±0.04	**0.63±0.03**	0.27±0.07	**0.62±0.05**	0.35±0.06	**0.57±0.01**
Micro P↑	0.42±0.07	**0.64±0.03**	0.38±0.05	**0.65±0.04**	0.40±0.05	**0.57±0.03**	0.36±0.06	**0.60±0.04**	0.35±0.05	**0.65±0.02**	0.39±0.04	**0.60±0.03**
Micro R↑	0.31±0.04	**0.53±0.05**	0.36±0.05	**0.58±0.06**	0.42±0.03	**0.58±0.07**	0.47±0.10	**0.70±0.08**	0.28±0.08	**0.64±0.07**	0.37±0.07	**0.58±0.04**
Micro F1↑	0.36±0.04	**0.58±0.02**	0.37±0.05	**0.61±0.02**	0.41±0.03	**0.57±0.03**	0.40±0.06	**0.64±0.04**	0.31±0.07	**0.64±0.04**	0.38±0.05	**0.59±0.02**
H Loss↓	0.37±0.03	**0.25±0.02**	0.41±0.03	**0.24±0.02**	0.40±0.02	**0.28±0.02**	0.46±0.04	**0.25±0.02**	0.40±0.04	**0.24±0.04**	0.40±0.03	**0.26±0.02**

Table 13 Comparative results for Scene dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.29±0.18	**0.31±0.20**	0.29±0.18	**0.31±0.20**	0.36±0.21	**0.38±0.23**	0.41±0.27	**0.42±0.28**	0.36±0.21	**0.37±0.22**	0.32±0.19	0.32±0.18
Recall↑	0.28±0.18	**0.30±0.20**	0.27±0.18	**0.29±0.20**	0.33±0.21	**0.35±0.22**	0.38±0.26	**0.39±0.27**	0.37±0.22	0.37±0.24	0.30±0.19	0.30±0.18
F1↑	0.28±0.18	**0.30±0.20**	0.28±0.18	**0.30±0.20**	0.34±0.21	**0.36±0.23**	0.39±0.26	**0.40±0.27**	0.36±0.21	0.36±0.22	0.30±0.19	**0.31±0.18**
Macro P↑	0.29±0.08	**0.30±0.09**	0.29±0.08	**0.30±0.11**	0.26±0.09	**0.28±0.09**	**0.28±0.10**	0.25±0.11	0.24±0.10	**0.25±0.10**	0.27±0.10	0.27±0.10
Macro R↑	0.22±0.08	0.22±0.07	0.21±0.07	**0.23±0.07**	0.27±0.11	**0.28±0.09**	0.30±0.12	0.30±0.12	0.28±0.09	0.28±0.09	0.25±0.09	0.25±0.09
Macro F1↑	0.20±0.07	**0.21±0.07**	0.20±0.06	**0.21±0.08**	0.21±0.06	**0.23±0.08**	0.24±0.09	0.24±0.09	0.22±0.07	**0.23±0.08**	0.21±0.06	0.21±0.07
Micro P↑	0.47±0.25	**0.49±0.25**	0.46±0.25	**0.48±0.26**	0.36±0.21	**0.38±0.23**	0.41±0.27	**0.42±0.28**	0.39±0.21	0.39±0.23	**0.43±0.23**	0.41±0.25
Micro R↑	0.28±0.18	**0.30±0.20**	0.28±0.17	**0.29±0.19**	0.33±0.19	**0.35±0.21**	0.38±0.24	**0.39±0.26**	0.38±0.22	0.38±0.23	0.30±0.18	0.31±0.18
Micro F1↑	0.35±0.21	0.37±0.22	0.34±0.20	**0.36±0.22**	0.35±0.20	**0.37±0.22**	0.39±0.26	**0.40±0.27**	0.38±0.21	0.38±0.23	0.35±0.20	0.35±0.21
H Loss↓	0.18±0.05	0.18±0.05	0.19±0.05	**0.18±0.06**	0.23±0.07	**0.22±0.08**	0.21±0.09	0.21±0.09	0.22±0.07	0.22±0.08	**0.20±0.06**	0.21±0.07

Table 14 Comparative results for Medical dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.24±0.06	**0.28±0.04**	0.27±0.05	**0.28±0.04**	**0.58±0.08**	0.56±0.08	0.38±0.06	0.38±0.07	0.43±0.05	**0.45±0.05**	0.25±0.03	**0.27±0.04**
Recall↑	0.21±0.06	**0.23±0.04**	0.23±0.05	0.23±0.04	0.52±0.07	0.52±0.07	0.31±0.04	0.31±0.05	0.41±0.04	**0.42±0.05**	0.22±0.03	0.22±0.04
F1↑	0.22±0.06	**0.24±0.04**	0.24±0.05	0.24±0.04	**0.54±0.07**	0.53±0.07	0.33±0.05	0.33±0.06	0.41±0.04	**0.43±0.05**	0.23±0.03	**0.24±0.04**
Macro P↑	0.08±0.03	**0.09±0.02**	0.09±0.03	0.09±0.02	**0.13±0.02**	0.10±0.01	0.03±0.01	0.03±0.01	**0.09±0.02**	0.08±0.01	**0.09±0.02**	0.08±0.02
Macro R↑	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	**0.12±0.01**	0.11±0.01	0.04±0.00	0.04±0.00	**0.10±0.02**	0.09±0.01	0.04±0.01	0.04±0.01
Macro F1↑	0.05±0.01	0.05±0.01	0.05±0.01	0.05±0.01	**0.11±0.02**	0.10±0.01	0.03±0.00	0.03±0.00	**0.09±0.02**	0.08±0.01	0.05±0.01	0.05±0.01
Micro P↑	0.79±0.10	**0.83±0.05**	**0.85±0.07**	0.81±0.06	**0.59±0.07**	0.58±0.08	0.38±0.06	0.38±0.07	0.57±0.09	0.57±0.07	**0.81±0.05**	0.78±0.08
Micro R↑	0.21±0.05	**0.24±0.04**	0.23±0.04	**0.24±0.04**	0.51±0.06	0.51±0.07	**0.31±0.04**	0.30±0.05	0.41±0.04	**0.42±0.04**	0.24±0.02	0.24±0.03
Micro F1↑	0.33±0.07	**0.37±0.05**	**0.37±0.05**	0.36±0.04	**0.55±0.06**	0.54±0.07	0.34±0.05	0.34±0.06	0.48±0.05	0.48±0.05	**0.37±0.03**	0.36±0.05
H Loss↓	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.03±0.00	0.03±0.00	0.03±0.00	**0.02±0.00**	0.02±0.00	0.02±0.00

Table 15 Comparative results for Cal500 dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.56±0.00	**0.57±0.01**	0.57±0.01	**0.58±0.01**	**0.35±0.01**	0.34±0.01	0.35±0.01	0.35±0.01	0.44±0.02	**0.46±0.01**	0.53±0.01	**0.54±0.01**
Recall↑	0.26±0.01	**0.27±0.01**	0.25±0.02	**0.27±0.00**	0.35±0.01	0.35±0.01	0.35±0.01	0.35±0.01	0.31±0.01	**0.32±0.00**	0.27±0.01	**0.28±0.01**
F1↑	0.35±0.01	**0.36±0.01**	0.34±0.02	**0.35±0.01**	**0.35±0.01**	0.34±0.01	0.34±0.01	0.34±0.01	0.36±0.01	**0.37±0.00**	0.34±0.01	**0.36±0.01**
Macro P↑	0.14±0.01	**0.16±0.02**	0.13±0.01	**0.16±0.02**	**0.17±0.01**	0.16±0.01	0.16±0.01	**0.17±0.01**	0.15±0.02	**0.17±0.00**	0.14±0.01	**0.17±0.01**
Macro R↑	0.08±0.00	0.08±0.01	0.08±0.01	**0.09±0.01**	**0.17±0.01**	0.16±0.01	0.16±0.01	**0.17±0.01**	0.12±0.01	**0.13±0.00**	0.09±0.00	**0.10±0.00**
Macro F1↑	0.09±0.00	**0.10±0.01**	0.08±0.01	**0.10±0.01**	0.16±0.01	0.16±0.01	0.15±0.01	**0.16±0.01**	0.13±0.01	**0.14±0.00**	0.10±0.00	**0.11±0.00**
Micro P↑	0.55±0.01	**0.56±0.01**	0.55±0.01	**0.56±0.02**	**0.35±0.01**	0.34±0.01	0.34±0.01	0.34±0.01	0.44±0.02	**0.45±0.01**	0.52±0.01	**0.53±0.01**
Micro R↑	0.26±0.01	**0.27±0.01**	0.24±0.02	**0.27±0.01**	**0.35±0.01**	0.34±0.01	0.35±0.01	0.35±0.01	0.31±0.01	**0.32±0.00**	0.26±0.01	**0.28±0.01**
Micro F1↑	0.35±0.01	0.36±0.01	0.34±0.02	**0.36±0.01**	**0.35±0.01**	0.34±0.01	0.34±0.01	0.35±0.01	0.36±0.01	**0.38±0.00**	0.35±0.01	**0.37±0.01**
H Loss↓	0.14±0.00	0.14±0.00	0.14±0.00	0.14±0.00	0.20±0.00	0.20±0.00	0.20±0.00	0.20±0.00	0.16±0.00	0.16±0.00	0.15±0.00	0.15±0.00

Table 16 Comparative results for Birds dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.09±0.03	0.09±0.03	0.09±0.04	0.09±0.04	0.10±0.04	0.10±0.05	**0.05±0.02**	0.04±0.01	**0.09±0.04**	0.08±0.03	0.10±0.05	**0.11±0.06**
Recall↑	0.07±0.02	0.06±0.02	0.06±0.03	**0.07±0.02**	**0.11±0.03**	0.10±0.03	0.03±0.01	0.02±0.01	0.05±0.01	0.05±0.01	0.08±0.04	0.08±0.04
F1↑	0.07±0.02	0.06±0.02	0.06±0.03	**0.07±0.03**	0.10±0.03	0.10±0.04	0.03±0.02	0.03±0.01	0.06±0.02	0.06±0.02	0.08±0.03	**0.09±0.04**
Macro P↑	**0.15±0.04**	0.12±0.04	0.13±0.04	**0.14±0.05**	0.14±0.03	0.14±0.05	**0.10±0.07**	0.07±0.04	**0.09±0.02**	0.08±0.02	0.12±0.04	**0.16±0.09**
Macro R↑	**0.08±0.03**	0.06±0.01	0.07±0.03	**0.08±0.03**	**0.13±0.04**	0.12±0.04	**0.05±0.03**	0.03±0.01	0.06±0.02	0.06±0.02	0.09±0.03	0.09±0.03
Macro F1↑	**0.09±0.02**	0.07±0.01	0.08±0.03	**0.09±0.03**	0.12±0.03	0.12±0.04	**0.06±0.04**	0.03±0.01	0.07±0.01	0.07±0.01	0.09±0.03	**0.10±0.04**
Micro P↑	0.34±0.08	0.34±0.10	0.30±0.08	**0.36±0.10**	0.23±0.05	**0.24±0.05**	**0.31±0.10**	0.26±0.05	**0.31±0.09**	0.27±0.05	0.30±0.07	**0.31±0.13**
Micro R↑	**0.13±0.05**	0.11±0.05	0.12±0.07	**0.13±0.06**	0.20±0.06	0.20±0.07	**0.07±0.03**	0.05±0.01	0.11±0.03	0.11±0.04	0.15±0.07	0.15±0.06
Micro F1↑	**0.19±0.06**	0.17±0.05	0.16±0.07	**0.19±0.07**	0.21±0.05	0.21±0.06	**0.11±0.04**	0.09±0.02	0.15±0.04	0.15±0.04	0.19±0.07	**0.20±0.08**
H Loss↓	0.06±0.01	0.06±0.01	0.06±0.01	0.06±0.01	0.08±0.01	0.08±0.01	0.06±0.01	0.06±0.01	0.06±0.00	0.07±0.01	0.07±0.01	0.07±0.00

Table 17 Comparative results for Enron dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.66±0.03	**0.68±0.03**	0.66±0.03	**0.68±0.03**	0.54±0.01	**0.56±0.01**	0.52±0.01	0.52±0.01	0.58±0.02	**0.59±0.02**	0.66±0.03	**0.67±0.03**
Recall↑	0.43±0.01	**0.45±0.01**	0.44±0.01	**0.46±0.01**	0.48±0.01	**0.50±0.02**	0.45±0.01	0.45±0.01	**0.46±0.02**	0.45±0.02	**0.46±0.01**	0.45±0.01
F1↑	0.49±0.01	**0.51±0.02**	0.50±0.02	**0.51±0.01**	0.49±0.01	**0.51±0.01**	0.46±0.01	0.46±0.01	0.48±0.02	**0.49±0.02**	**0.52±0.01**	0.51±0.01
Macro P↑	0.20±0.02	**0.21±0.02**	0.20±0.02	0.20±0.02	0.21±0.01	**0.22±0.01**	**0.09±0.01**	0.08±0.01	0.19±0.01	0.19±0.01	0.21±0.02	**0.22±0.02**
Macro R↑	0.10±0.01	0.10±0.01	0.10±0.01	0.10±0.01	0.13±0.01	0.13±0.02	0.08±0.00	0.08±0.00	**0.13±0.01**	0.12±0.01	0.11±0.01	0.11±0.01
Macro F1↑	0.12±0.01	0.12±0.01	0.12±0.01	0.12±0.01	0.14±0.01	**0.15±0.02**	0.07±0.00	0.07±0.00	0.14±0.01	0.14±0.01	0.12±0.01	0.12±0.01
Micro P↑	0.71±0.01	0.71±0.02	0.70±0.02	**0.71±0.02**	0.56±0.01	0.56±0.01	0.54±0.01	0.54±0.01	0.59±0.01	**0.61±0.01**	0.69±0.02	**0.71±0.02**
Micro R↑	0.42±0.01	**0.43±0.01**	0.43±0.01	**0.44±0.01**	0.44±0.01	**0.46±0.02**	0.40±0.01	0.40±0.01	0.44±0.02	0.44±0.02	**0.45±0.01**	0.43±0.02
Micro F1↑	0.53±0.01	**0.54±0.01**	0.53±0.02	**0.54±0.01**	0.49±0.01	**0.50±0.01**	0.46±0.01	0.46±0.01	0.51±0.01	0.51±0.01	0.54±0.01	0.54±0.02
H Loss↓	0.05±0.00	0.05±0.00	0.05±0.00	0.05±0.00	0.06±0.00	0.06±0.00	0.06±0.00	0.06±0.00	0.05±0.00	0.05±0.00	0.05±0.00	0.05±0.00

Table 18 Comparative results for Foodtruck dataset.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.57±0.08	**0.59±0.11**	0.55±0.10	**0.63±0.10**	0.54±0.12	**0.67±0.12**	0.67±0.14	0.67±0.14	0.45±0.09	**0.52±0.08**	0.59±0.11	**0.63±0.12**
Recall↑	0.44±0.09	0.44±0.05	0.39±0.09	**0.41±0.06**	0.38±0.08	**0.43±0.07**	0.41±0.09	0.41±0.09	0.43±0.04	**0.47±0.05**	0.43±0.08	**0.46±0.08**
F1↑	0.44±0.07	**0.45±0.06**	0.40±0.08	**0.45±0.07**	0.39±0.08	**0.48±0.08**	0.47±0.10	0.47±0.10	0.39±0.05	**0.43±0.06**	0.44±0.07	**0.48±0.09**
Macro P↑	0.19±0.04	**0.20±0.03**	0.15±0.03	**0.19±0.03**	0.16±0.02	**0.19±0.03**	0.06±0.01	0.06±0.01	0.16±0.01	**0.19±0.03**	0.16±0.05	**0.19±0.06**
Macro R↑	0.13±0.01	**0.14±0.01**	0.11±0.01	**0.12±0.01**	**0.13±0.02**	0.12±0.02	0.08±0.00	0.08±0.00	0.14±0.02	**0.18±0.02**	0.12±0.01	**0.14±0.02**
Macro F1↑	0.14±0.02	0.14±0.01	0.11±0.02	**0.12±0.02**	0.13±0.02	0.13±0.03	0.07±0.01	0.07±0.01	0.14±0.01	**0.17±0.02**	0.12±0.02	**0.14±0.03**
Micro P↑	0.60±0.10	**0.65±0.10**	0.59±0.10	**0.67±0.12**	0.49±0.08	**0.65±0.11**	0.67±0.14	0.67±0.14	0.48±0.04	**0.51±0.04**	0.57±0.09	**0.66±0.13**
Micro R↑	0.34±0.05	**0.36±0.05**	0.31±0.04	**0.33±0.06**	0.31±0.05	**0.33±0.06**	0.30±0.07	0.30±0.07	0.35±0.04	**0.40±0.05**	0.34±0.04	**0.36±0.07**
Micro F1↑	0.43±0.06	**0.46±0.07**	0.40±0.05	**0.44±0.08**	0.38±0.06	**0.44±0.08**	0.42±0.09	0.42±0.09	0.40±0.04	**0.44±0.04**	0.43±0.05	**0.46±0.09**
H Loss↓	0.17±0.03	**0.16±0.02**	0.17±0.03	**0.16±0.03**	0.19±0.03	**0.16±0.03**	0.16±0.03	0.16±0.03	0.19±0.02	0.19±0.02	0.17±0.02	**0.16±0.03**

Table 19 Comparative results for Yeast dataset between Native and TEN.

Metric	BR		CC		LP		MLTSVM		ML-*K*NN		RAkELd	
	Native	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**
Precision↑	0.73±0.03	**0.85±0.03**	0.73±0.03	**0.83±0.02**	0.66±0.02	**0.79±0.03**	0.67±0.02	**0.83±0.03**	0.64±0.02	**0.78±0.03**	0.70±0.03	**0.82±0.02**
Recall↑	0.55±0.01	**0.76±0.00**	0.57±0.02	**0.76±0.02**	0.62±0.02	**0.74±0.01**	0.63±0.01	**0.77±0.01**	0.60±0.02	**0.75±0.01**	0.61±0.04	**0.76±0.00**
F1↑	0.60±0.01	**0.78±0.01**	0.61±0.02	**0.77±0.02**	0.62±0.02	**0.74±0.02**	0.63±0.02	**0.78±0.02**	0.59±0.02	**0.74±0.01**	0.62±0.02	**0.77±0.01**
Macro P↑	0.57±0.05	**0.75±0.04**	0.56±0.07	**0.73±0.04**	0.49±0.06	**0.66±0.03**	0.50±0.04	**0.71±0.02**	0.47±0.02	**0.65±0.04**	0.51±0.03	**0.72±0.04**
Macro R↑	0.32±0.00	**0.54±0.01**	0.34±0.01	**0.53±0.02**	0.39±0.01	**0.54±0.01**	0.38±0.00	**0.54±0.01**	0.40±0.01	**0.59±0.02**	0.36±0.03	**0.54±0.01**
Macro F1↑	0.35±0.00	**0.59±0.01**	0.37±0.01	**0.58±0.02**	0.40±0.01	**0.57±0.01**	0.37±0.00	**0.58±0.01**	0.42±0.01	**0.61±0.02**	0.37±0.01	**0.58±0.01**
Micro P↑	0.74±0.03	**0.87±0.03**	0.73±0.02	**0.85±0.02**	0.67±0.02	**0.79±0.02**	0.68±0.02	**0.84±0.03**	0.65±0.02	**0.78±0.03**	0.70±0.03	**0.83±0.02**
Micro R↑	0.55±0.01	**0.75±0.01**	0.57±0.01	**0.74±0.02**	0.61±0.02	**0.73±0.01**	0.62±0.01	**0.75±0.01**	0.59±0.01	**0.74±0.01**	0.60±0.03	**0.75±0.01**
Micro F1↑	0.63±0.01	**0.80±0.01**	0.64±0.01	**0.79±0.01**	0.64±0.02	**0.76±0.01**	0.65±0.01	**0.79±0.01**	0.62±0.01	**0.76±0.01**	0.65±0.02	**0.79±0.01**
H Loss↓	0.19±0.01	**0.11±0.00**	0.19±0.01	**0.12±0.01**	0.21±0.01	**0.14±0.00**	0.20±0.01	**0.12±0.01**	0.22±0.01	**0.14±0.01**	0.20±0.01	**0.12±0.00**

To explore the sensitivity of the proposed technique, we experimented with an experiment using TEN as the feature reconstruction method and compared it to the Native (original data features). We experimented using the same set of six classifiers, and the result is shown in Table 19.

Table 19 shows the performance of the proposed TEN in construct a new feature set and compares the results of the classification obtained from the native data features. We can observe that the proposed TEN is superior to the native features when they are classified by the six classification techniques ($p = 0.0001$). In addition, the visual representation of the performance comparison is shown in Figures 10 - 15.

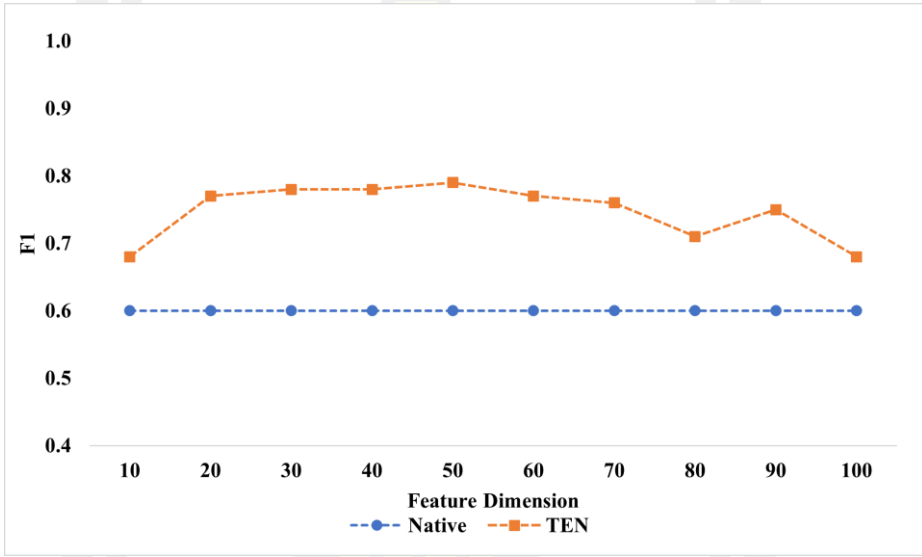


Figure 9 The results were obtained from the proposed feature reconstruction method and the native data feature.

Figure 9. shows the results of the classification of the proposed technique (TEN) and the native data features when the size of the reconstructed is varied, from m' =10 to m' =100. It can be observed that TEN gives better results than the native features, even if the dimension of the reconstructed feature is small (m'=10). Figures 10 – 15. depicts the comparative representation of different evaluation metrics.

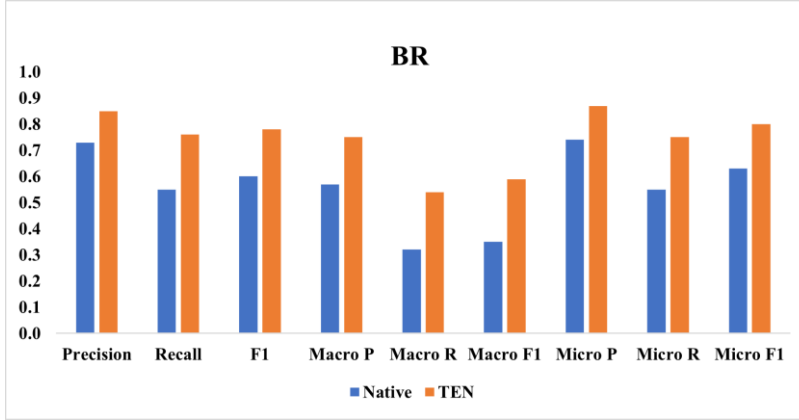


Figure 10 The BR measurement results compare the native feature with TEN.

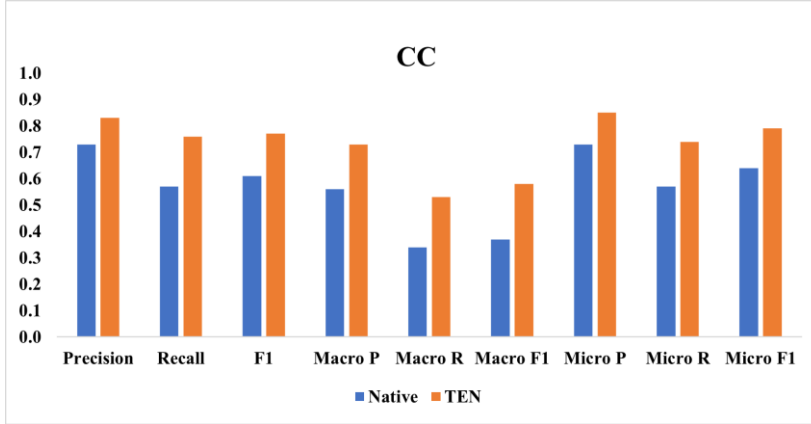


Figure 11 The CC measurement results compare the native feature with TEN.

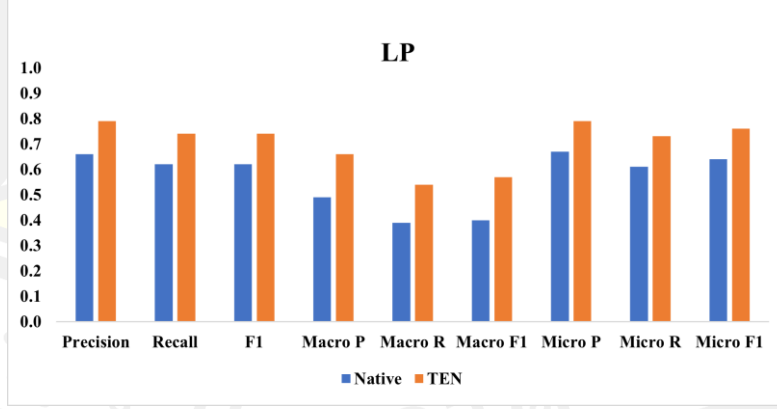


Figure 12 The LP measurement results compare the native feature with TEN.

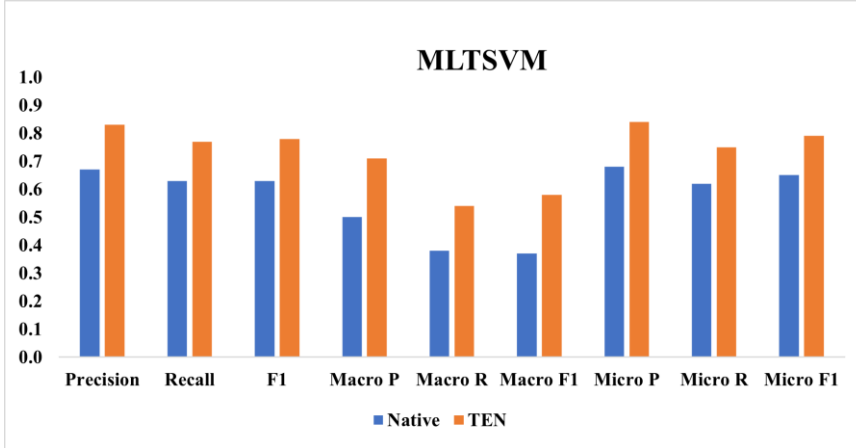


Figure 13 The MLTSVM measurement results compare the native feature with TEN.

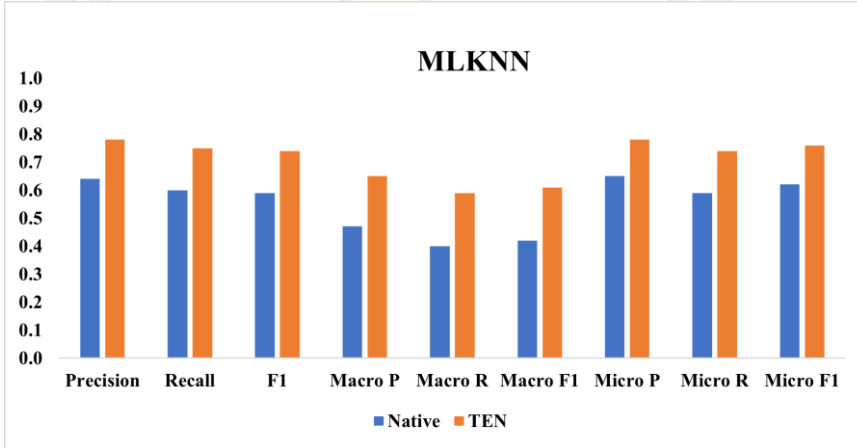


Figure 14 The ML-KNN measurement results compare the native feature with TEN.

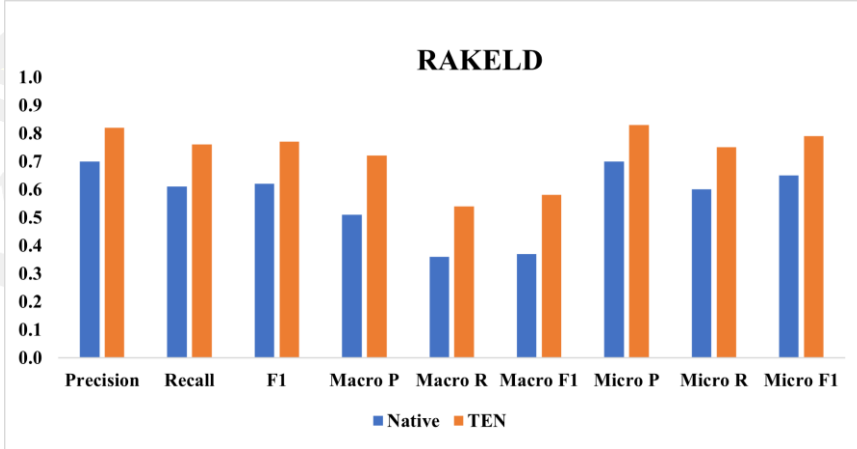


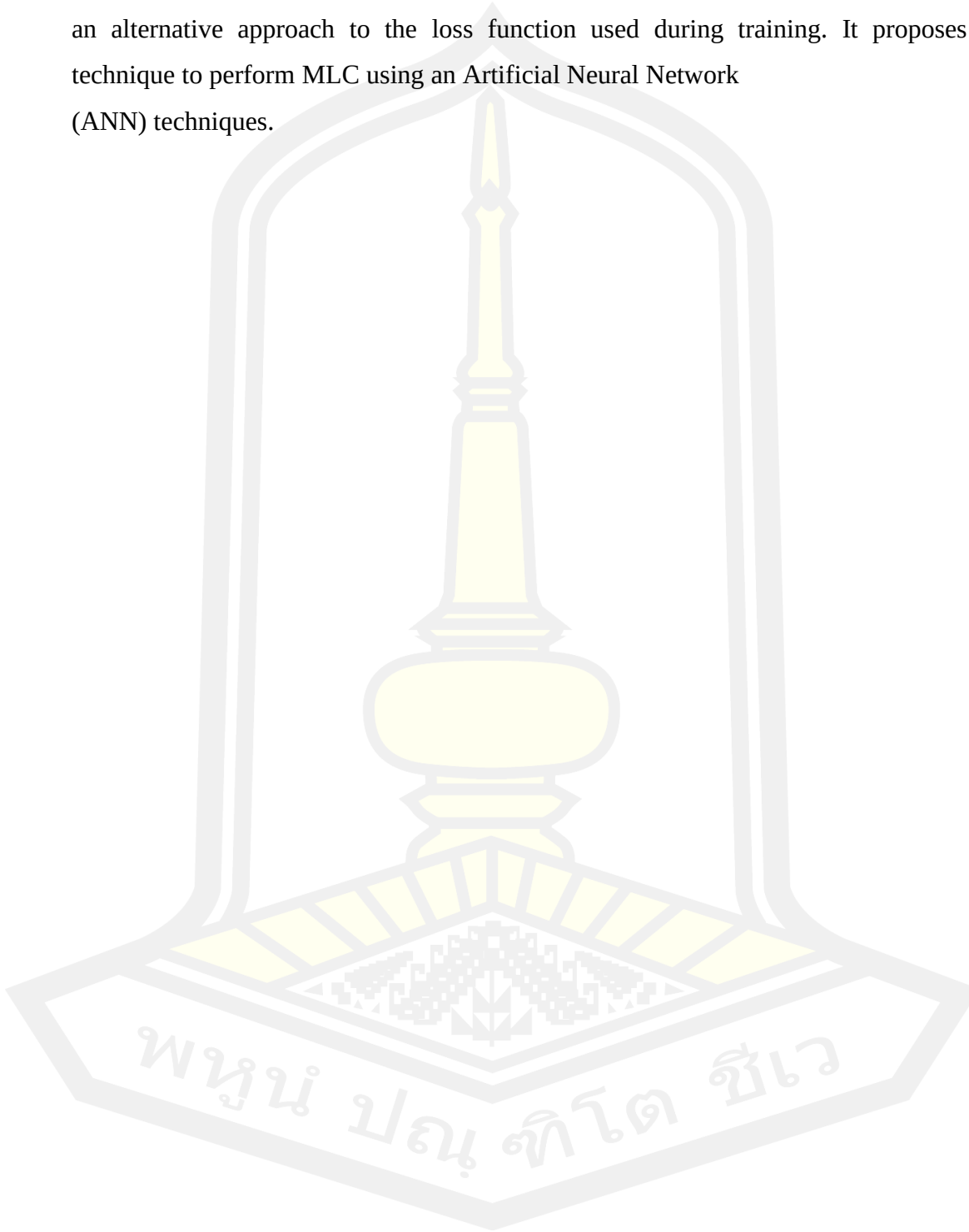
Figure 15 The RAkELd measurement results compare the native feature with TEN.

3.6 Conclusion

In this study, we propose a technique for improving the performance of MLC with a feature reconstruction method. The proposed feature reconstruction applied the AutoEncoder technique that intentionally encodes the input data instance to generate a compact feature representation of them. We implemented two of the construction procedures. AutoEncoder alone (EN) was built to encode the feature subsets of the data instances. The TEN was constructed to derive a compact set of the data instances and maintain the contextual insights of the dataset, conveying the class-label representation. To evaluate the performance of the proposed method, we collected 8-standard datasets, which are acquired from different domains and different data settings. We conducted the experiments by applying six classifiers, which are derived from three different MLC techniques (i.e., PTM, AM, and EM). The experiments were separated into two folds. The first experiment explored the effectiveness of the TEN and EN in the feature reconstruction process. In comparison, the second experiment was objected to measuring the proposed technique's performance compared with the original data feature used in MLC.

The experimental results, presented in Section 3.5, deliberately delineate the performance of the proposed technique. For all data sets, TEN essentially provides promising results, which is better than EN. TEN works well with the Yeast and Emotion dataset, giving better results for all the MLC algorithms and the measurement metrics. The Yeast and Emotion are the only two datasets with high density shown in Table 10. The density of the dataset in MLC indicates the well-presentation of the class labels. Therefore, TEN trends work well with the high-density dataset for MLC problems. In addition, the results obtained from the second experiment on the Yeast dataset show that the reconstruction technique is superior to the native data features. In general, feature reconstruction can produce different sizes of compact features. Therefore, we varied the sizes of the reconstructed features to observe the sensitivity of the technique. The results indicate that TEN gives better results than the native features for all MLC problems and measurement metrics.

In the next chapter, we will investigate to improve the performance of the classification problems, especially on datasets with low density. The work examines an alternative approach to the loss function used during training. It proposes a technique to perform MLC using an Artificial Neural Network
(ANN) techniques.



Chapter 4

A Comparative Study of Applying Neural Network-based Techniques for Solving Multi-label Classification Problems

4.1 Introduction

Multi-label classification (MLC) is one of the supervised learning methods that has increasingly gained attention in the research into solving classification problems (Chandran & Panicker, 2017; Prajapati & Thakkar, 2021). MLC is a challenging problem of classification as the technique explicitly classifies data instances into a set of mutual classes. For example, classifying patients with multiple diseases (Sangkatip & Phuboon-Ob, 2020), detecting images with multiple objects, video clips are in several categories, and classification sounds in different emotions. Thus, general multi-class classification (MCC) cannot be coupled with these classification problems. The MLC method is an approach to solving MLC problems and the MLC method was presented in 2004 by Boutell et al. (2004) to present the classification of objects in a multi-object image. Then, the research into MLC has intensively been one of the areas that interest the researchers across machine learning applications. Initially, general approaches to solving MLC problems can be categorized into different based methods. Tsoumakas & Katakis. (2007b) divided MLC techniques into two groups: Adaptation Methods (AM) and Problem Transformation Methods (PTM). The AM method applies and adapts MCC techniques for perform the MLC process. There are a number of existing MCC techniques that have been implemented to these problems, for example, Decision Tree algorithm C4.5, which is called ML-C4.5 (Clare & King, 2001), and the K-Nearest Neighbors algorithm, so-called ML-*K*NN (M. L. Zhang & Zhou, 2007). The PTM method essentially converts MLC problems to multi-class problems. Therefore, the conventional MCC classification algorithm can be applied to the problems accordingly. The techniques in this classification family are, for instance, Binary Relevance (BR) (Godbole & Sarawagi, 2004), Classifier Chains (CC) (Read et al.,

2011a), Label Power-set (LP). Madjarov et al. (2012b) presented a technique and conducted experiments to compare the effectiveness of MLC methods. Then, the MLC method was divided into three groups, retaining the same AM and PTM methods. A new group called Ensemble Methods (EM) was introduced, which was developed by combining PTM methods to improve classification efficiency, such as RAkEL (Tsoumakas & Vlahavas, 2007b), EPS (Read et al., 2008) and ECC (Read et al., 2011a).

The current MLC challenges are focused on improving classification efficiency with greater accuracy. More research has been done to analyze feature-label relationships. Feature Engineering (FE) (Guozhu & Huan, 2018; Hafeez et al., 2021) is divided into several tasks, for example, feature selection (FS), feature transformation (FT), and feature reconstruction (FR). Dimensional reduction is one of the techniques used to transform data features. There are two categories of dimensionality reduction methods. One is feature selection and feature transformation. Method FS keeps only useful features and dismisses others, while FT constructs a new but smaller number of features out of the original ones (Deng et al., 2013a). The current FT method can be applied by implementing, for example, deep learning algorithms (Patterson & Gibson, 2017) and unsupervised network algorithms, which learn to encode data to extract the relationships of the data. Y. Cheng et al. (2019) used a deep learning technique to build and extract relationships between attributes and labels in a multi-label classification. Feature reconstruction, a transformation process, can be considered as a tool to generate a set of new feature sets (based on the original data features). The reconstructed features are anticipated to be compact and descriptive, which can be used in the classification process.

The correlation of the labels in the data is one of the statistical-based techniques that have been applied to the problems, such as Label Correlation (LC) (J. Li et al., 2022; Nazmi et al., 2021; Xiao et al., 2021). The method examined the label relationships of the data, which assumed that the resulting labels were interdependent among the labels. Therefore, this method determined the probabilities of a label from the relation of the data with respect to the label. M.-L. Zhang, (2011) proposed the LIFT, which adapted the 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 established by creating additional attributes of the data (Gao et al., 2020). Huang et al. (2018) proposed a technique to learn the dispersion of label attributes, including common attributes. They applied double-label correlation to differentiate labels for each category.

In the past decades, Neural Network approaches have been among the techniques applied to solve generics machine learning problems and applications, including multi-label classification. Several studies have shown that Neural Network methods can improve classification performance. X. Zhang et al. (2019) proposed a method using a Deep Neural Network (DNN) to classify multi-label data called GroupNet. GroupNet used a Convolutional Neural Network (CNN) to perform that classification (Valueva et al., 2020). They constructed a single dimension architecture applicable to the dataset for the classification. Lipton et al. (2016) proposed Recurrent Neural Networks (RNN) along with a number of research such as (S.-F. Chen et al., 2017; Nápoles et al., 2021). They applied RNN to perform the classification tasks. Maxwell et al. (2017) proposed a deep learning architecture for multi-label classification. DNN is also a wide range techniques that recently have been implemented for the problems (Lian et al., 2019; Yeh et al., 2017). The techniques have potentially proved to be generic to improve classification accuracy. Therefore, in this work, the proposed method will rely on Artificial Neural Network (ANN) technique. The work will essentially construct an ANN to perform the classification task. In addition, this work will integrate the information of the label patterns in the data to enforce the generalization of the model. The classification will account for this information to predict the classification result toward the patterns of a label, which is exhibited in the data.

The paper is organized as follows: Section 4.2 provides an explanation of neural networks for multi-label classification. In Section 4.3, the proposed materials and methods are described. The datasets used in this work are also delineated and explored. Section 4.4 demonstrates that experiments and results were conducted to evaluate the performance of the proposed before the discussion of this work given in Section 4.5., Finally, the conclusion.

4.2 Neural Network for Multi-label Classification

4.2.1 Preliminaries

Let X be a space of data instances comprising n data instances x, i.e. $\forall x \in X, x = x_1, \dots x_d$ (where d is the number of instance features) a set of d-dimensional features divided from x , and a set p a possible label space $Y = y_1, \dots y_p$, 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 set of training networks $N = n_1, .. n_k$ is a feasible solution for neural networks. Then, the objective is to search for an optimal n such that the predicted outcomes Y' meet the majority of the true value of Y. To construct a generalized model for the task, the network takes the input x through a set of hidden layers $H = h_1(.), \dots, h_L(.)$ where L is the number of the hidden layer in the network. Therefore, the prediction obtained from a network can be evaluated by the following:

$$[y'_1, \dots y'_m] = h_1(x), .., h_L(x), \tag{22}$$

then we can simplify to

$$\mathrm{y}' = H(x) \tag{23}$$

Each hidden layer contains a set of adjustable and trackable parameters $\boldsymbol{w}$ and $\boldsymbol{b}$. In addition, the last layer of the network is augmented by activation functions that activate the outputs from the last layer to the prediction outcomes.

4.2.2 Constructing Neural Network for the Classification

The network architecture has d inputs (including a bias term b) 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 can consist of multiple different layers stacked together, but it is generally assumed that the performance of an MLP does not increase past two layers. The hidden layer is connected to the output layer, where the output layer is the same number of classes that are getting predicted. The calculation above happens for each node in the network until the output layer is reached. At this point, called a forward

pass, the network has tried to learn about the sample passed in and has made a prediction about that data, where the nodes of the output layer are probabilities that the sample is of a certain class. This is the point where backpropagation takes over. This process of a forward pass and backpropagation continues until a certain number of iterations are met or the network converges on an answer. Another way to look at the method is that the architecture uses the data to find a mathematical model or function to best describe the data. As the network is trying to learn, it is constantly searching for a global minimum value such that predictions can be accurate.

4.2.3 Backpropagation for Multi-Label Learning (BP-MLL)

In MLC each data point is associated with a set of labels whose size is not fixed. The number of all possible labels is known, but not the size of the label set associated with each data point. An example of such a type of problem could be keyword extraction for news articles. A news article can be associated with several possible keywords in the keyword universe, and the number of such keywords may not be known apriori. A simple way of solving such a problem would be to devise a binary classification problem by training n independent binary classifiers, each predicting whether a news article belongs to a certain class (has a certain keyword). One of the downsides of such a method is that n separate models need to be trained, which is computationally expensive both during training and inference. The other downside of such an approach is that correlation information between labels is completely ignored, which may not be desirable in most cases.

Turns out a simple modification of loss function while training a neural network allows us to learn multilabel classification efficiently.

$$E = \sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i \times \bar{Y}_i} exp(-(c_k^i - c_l^i) \quad (24)$$

This loss function considers that relation of a pair of labels $(k, l) \in Y_i \times \bar{Y}_i$, where k is relevant to the data instance x_i and l is not, if the prediction score for k is positive whereas the prediction score for 1 is negative, then $exp(-(c_k^i - c_l^i)$ has the minimum penalty. An incorrect prediction score order results in a higher penalty. Therefore, minimizing the equation would result in pairs of labels being predicted correctly.

4.3 Materials and Methods

The main objective of this work is to develop a technique that can classify multi label data efferently. This work integrates the information of the patterns of labels exhibited in the data. The pattern information will be used to implement additional loss terms (so called Soft-loss). The overall process of the technique presented in this work is illustrated in Figure 16.

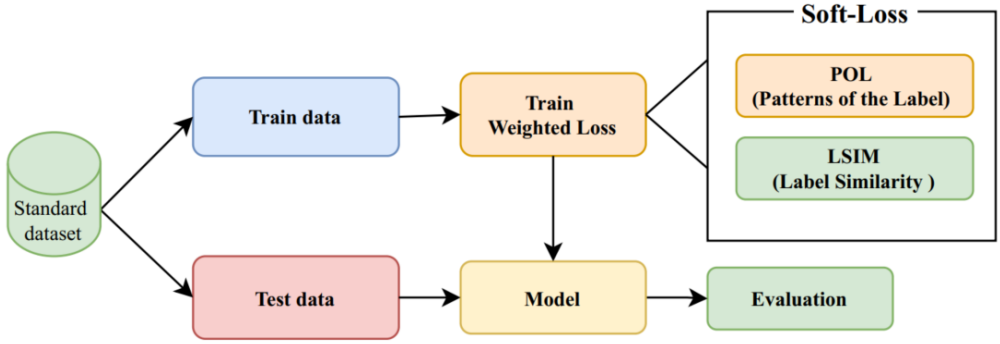


Figure 16 The overall process of the proposed method.

The dataset is divided into two folds, train data and test data. The patterns of labels are invested by examining all the cardinality of the label Y in the data. These patterns of the label will be used to construct the Soft-loss. The Soft-loss will be weighted and tuned toward the optimal solution. The Soft-loss anticipated the training to converge and direct the solution to the existing patterns in the data.

Therefore, this section will provide the details of the data used in this work. Then, the patterns of the labels of the data will be explored and analyzed. The

distribution of the label patterns over the dataset will be illustrated. Finally, the methodology of incorporating the Soft-loss will be explained.

4.3.1 Dataset Pattern Analysis

The datasets were collected from the Mulan datasets (Tsoumakas, Spyromitros-Xioufis, et al., 2011). The data contains 8-datasets with different data topologies, depending on the domains. In addition, each dataset has different characteristics and properties, such as the number of data instances, the number of feature dimensions, Cardinality, and Density. The summary of the dataset is shown in Table 20.

Table 20 The details of multi-label datasets used in this work.

Datasets	Domain	Instances	Features	Labels	Cardinality	Density
birds	audio	645	260	19	1.014	0.053
enron	text	1702	1001	53	3.378	0.064
emotions	music	593	72	6	1.869	0.311
medical	text	978	1449	45	1.245	0.028
yeast	biology	2417	103	14	4.237	0.303
scene	image	2407	294	6	1.074	0.179
cal500	music	502	68	174	26.044	0.15
foodtruck	recommend	407	21	12	2.29	0.191

In general, the Cardinality is the average number of labels per example, defined in Equation 1.

The Density is the number of labels per sample divided by the total number of labels, defined in Equation 2.

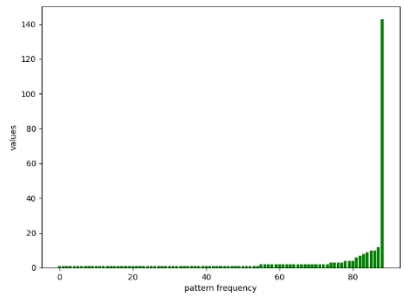
From our inspection, each of the data instances in the dataset is assigned to a particular multi-label entity (Y). We analyze the patterns of labels from the training dataset in order to examine the possible patterns of data labels in the dataset. To extract the patterns of the dataset, we iterate all the data instances and their associated data labels. The patterns of the labels in the data are one of the critical pieces of information that can be used to improve the classification by constraining the model to essentially predict the classification output to the existing label patterns

exhibiting the datasets. Therefore, the permutation of data labels or patterns is illustrated in Table 21.

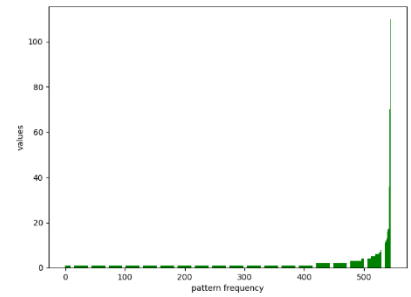
Table 21 The number of label patterns of the training dataset used in this work.

Datasets	**No. of Label Patterns**			
	Train set	**No. Instances > 0**	**No. Instances > 10**	**No. Instances > 10 (%)**
birds	322	89	2	48.14
enron	1123	545	10	27.96
emotions	391	26	11	87.21
medical	333	61	9	64.26
yeast	1500	164	26	74.60
scene	1211	14	9	98.76
cal500	401	401	-	-
foodtruck	325	101	6	52.30

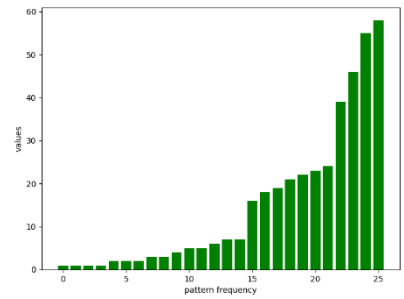
The pattern of the labeled dataset is the number of patterns embedded in the dataset. It can then display the number of labels occurring for each pattern. For example, the yeast dataset contains 164 pattern labels. Then analyze the number of each pattern to witness how many there are. A frequency graph of each data set can be displayed, as shown in Figure 17.



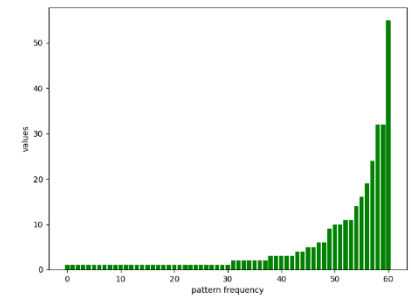
(a) Birds



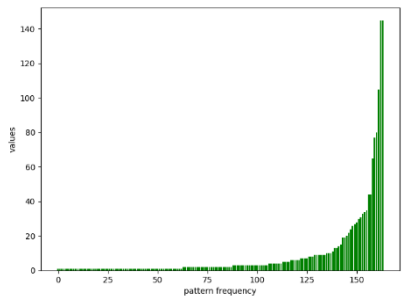
(b) Enron



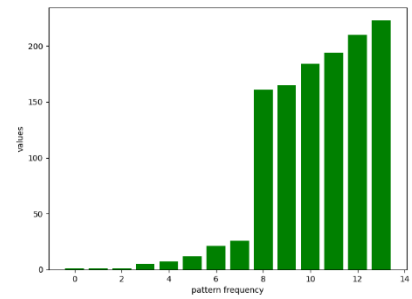
(c) Emotions



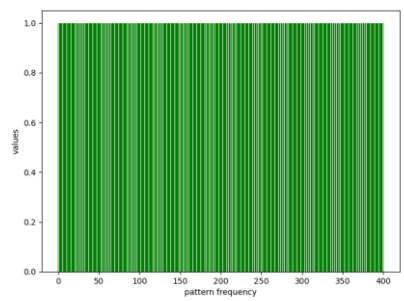
(d) Medical



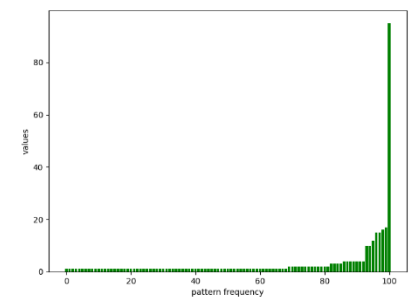
(e) Yeast



(f) Scene



(g) Cal500



(h) Foodtruck

Figure 17 The distribution of the label patterns in each of the datasets.

Table 22 Pattern of the predicted test dataset.

Datasets	Test	No. of Patterns > 10	No. of Patterns (%)
birds	323	2	52.94
enron	579	3	30.56
emotions	202	7	74.25
medical	645	15	75.81
yeast	917	19	68.26
scene	1196	9	98.24
cal500	101	-	-
foodtruck	82	45	54.87

4.3.2 Method

Backpropagation for Multi-Label Learning (BP-MLL) (Mandziuk & Zychowski, 2019; M. L. Zhang & Zhou, 2006). This feed-forward neural network for MLC problems uses an error function to capture the correlation among the labels. This function penalizes the predictions that include labels that are not truly relevant to the processed instance.

Multi-label Hierarchical Adaptive Resonance Associative Map Neural Network (ML-HARAM) (Benites & Sapozhnikova, 2016, 2017). This neural system was initially developed for text datasets with high dimensionality. Overall, it aims to increase the classification speed by adding an extra layer of adaptive resonance theory to group the learned prototypes into large clusters.

This work proposes two classification techniques, i.e., (i) Patterns of the Label (POL) and (ii) Label Similarity (LSIM). POL assumes that the prediction of the class label will be according to the existing patterns of the labels in the data (shown in Table 21 and Table 22). Therefore, POL utilizes the information of the label patterns in the classification process. POL starts by examining the patterns of the labels $P = \{p_1, .. p_n\}$ when n is the number of patterns in the datasets. For each pattern p, there exists the outer candidate label (p_d), which is determined by the center of p_c. Then, given P, we can construct a set outer candidate label $Y_p = \{y_{p1}, .. y_p\}$. Finally, the loss function of the model can be calculated by using Y_p. The

weighted loss term is separated into two parts, (i) native term and (ii) pattern term, which is demonstrated in Algorithm 1.

Algorithm 1 The computational algorithm of the POL method

Input: y, y', y_p, α
Output: L
1: $\epsilon \leftarrow 0$
2: **for** $i \leftarrow 1$ to N **do**
3: $\quad l_1 \leftarrow -\frac{1}{|y_i|} \sum_i (y_i' \log(y_i) + (1 - y_i') \log(1 - y_i))$
4: $\quad l_2 \leftarrow f_d(y', y_p)$
5: $\quad \epsilon \leftarrow l + (\alpha * l_1) + (1 - \alpha * l_2)$
6: **end for**
7: $L = \frac{1}{N} \times \epsilon$
8: **return** L

where f_d denotes a distance function that determines the spatial differences between the prediction (y') and the expected centriod of the train pattern (y_p), which is define as follows:

$$f_d(y', y_p) = \sqrt{\sum_{|y'|} (y' - y_P)^2} \tag{25}$$

and $\alpha = [0,1]$ is the weight of the loss terms.

Similarly, LSIM is introduced as the information that can be used to generalize the model. With LSIM, data instances classified into the same label pattern are supposed to account for a similar feature composition. On the other hand, the patterns of labels can eventually relate to the feature. For each training process, LSIM examines the similarity of the data instance (features). Therefore, 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 the *K*NN 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
$\epsilon \leftarrow 0$
for $i \leftarrow 1$ to N **do**
  $k \leftarrow 0$
  for $j \leftarrow 1$ to G **do**
    $k \leftarrow -\frac{1}{|y_{ij}|} \sum_i (y'_{ij} \log(y_{ij}) + (1 - y'_{ij}) \log(1 - y_{ij}))$
  end for
  $\epsilon \leftarrow (\frac{k}{|G|})$
end for
$L = \frac{1}{N} \times \epsilon$
return L

In the computational analysis of the proposed algorithms with Big-O notation, the POL and LSIM algorithms optimize the training process of the neural network to learn the pattern of labels. The neural network algorithm feeds h hidden neurons with n inputs each; hence $h * O(n)$ plus feeds m output neurons with h inputs each, $m * O(h)$. It is defined the following notation : $h * O(n) + m * O(h) = O((n + m) * h)$.

The LSIM examines the similarity of the data instance. A set of similar instances S where k is the number of similar members. S is generated by the *K*NN algorithm. Given S, that there exists a label set G associated with S. Then, the loss term of the neuron network is calculated over the average loss in G and the predicted label. It is defined the following Big-o notation is $O((n + m) * h) + O(n^3)$.

The POL utilizes the information of the label patterns in the classification process. POL starts by examining the patterns of the labels P when n is the number of patterns in the datasets. For each pattern p, there exists the outer candidate label (p_d), which is determined by the center of p_c. Then, given P, we can construct a set outer candidate label Y_p. Finally, the loss function of the model can be calculated by using Y_p. The weighted loss term is separated into two parts, (i) native term and (ii) pattern term. It is defined the following Big-o notation is $(O((n + m) * h) * O(n)) + O(n^2)$.

4.4 Experiments and Results

4.4.1 Experiment setup

This work proposes two methods (POL and LSIM) for the MLC problems, which were explained in the previous section. In this section, we will demonstrate the experiments and the results from the proposed techniques and the state-of-the-art method with respect to the Neural Network-based methods, which are BP-MLL and ML-HARAM. This work uses eight datasets, according to Table 20. The feature's size in the dataset ranges between 21 and 1449 dimensions. In addition, the length of the label is between 6 and 174 accordingly. The setting up of the experiments is as follows.

The BP-MLL (M. L. Zhang & Zhou, 2006) method is a feed-forward Neural Network method that was used in the experiments. In this method, a loss function is defined using the cross-entropy scheme. We defined the parameters used in the experiment as follows: the input layer was set along with the size of the features (attributes), the number of hidden layers was set to two layers for simplicity, and the output layer was equal to the number of labels. We used ReLU as the activation of the input layer and hidden layer and utilized the sigmoid function for the output layer. The training was carried out in a total of 10 epochs.

The ML-HARAM (Benites & Sapozhnikova, 2016, 2017) method aims at increasing the classification speed by adding an extra ART layer for clustering learned prototypes into large clusters. We defined the parameters used in the experiment as follows: set vigilance to 0.95 as parameters for adaptive resonance theory networks and defined the threshold value as 0.05, which controls how many prototypes participate in the prediction.

The ANN method was also used in the experiments for comparison purposes. We defined the parameters used in the experiment as follows: the input layer was equal to the size of the attributes, the number of hidden layers designated to two layers, and the output layer was set to the number of labels. ReLU was used for the activation of the input layer and hidden layer and Sigmoid activation for the output layer and loss function set to binary cross-entropy. Asam optimizer was applied with the training of a total 10 epochs.

The POL is the proposed method of this study that customized the loss function of the training process. This loss determines the variability of the label patterns in the dataset. The first loss calculates the predicted value and the actual value of the pattern formed by the label. We explore the number of label patterns that are exhibited in the data (the member instances of each pattern > 1. , 10, 50, 70). The second loss calculates the Binary Cross-Entropy between predicted and actual, then averages the two losses assigned alpha to weight the two losses and the network parameter. Then, training was carried out in a total of 10 epochs.

The LSIM is also proposed in this research. The technique designates the loss function by 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 class actual values and label of the nearest data instance was used to obtain the model loss. In this work, we set k = 3 as the number of similar data instances. Finally, the loss was obtained by averaging all the losses of each data pair.

The experiments have implemented the algorithm of Scikit-multilearns (Szymański & Kajdanowicz, 2017), which is the library for MLC built on top of the well-known. All experiments are carried out on a Core i7 (1.99 GHz) on Windows 10 machine with 20.0 GB RAM.

In the experiment, we utilized Scikit-multilearn as the main tool to conduct various experiments (Szymański & Kajdanowicz, 2017). We choose ten common evaluation metrics for MLC (Wu & Zhou, 2017). These evaluation metrics cover both example-based metrics and label-based metrics, namely, Precision, Recall, F1, Macro Precision (Macro P), Macro Recall (Macro R), Macro F1, Micro Precision (Micro P), Micro Recall (Micro R), Micro F1, and Hamming Loss (H Loss). The measurement metrics are defined in Equations 12 – 21.

For each classifier, true positives (tp_j), true negatives (tn_j), false positives (fp_j), and false negatives (fn_j) obtained (based on the metrics) are calculated – for each label $y: j = 1 \ldots m,$. Macro F_1 is essentially the harmonic mean obtained from Precision and Recall based on an average of each label y_j and an average of overall labels. In addition, Micro F_1 is the harmonic mean of Micro derived from Precision and Micro Recall in the above definition.

4.4.2 Experiment Results

After setting out the experiment, we carried out the experiment with eight datasets. We evaluate the performance of the classification technique, the experiments were conducted using BP-MLL, ML-HARAM, ANN, LSIM, and POL, and the results are demonstrated in Tables 23 – 32. For each evaluation metric, "↑" indicated "the larger, the better" while "↓" indicated "the smaller, the better". Each row of datasets contains the results of five algorithms in the column. The best algorithm results are enclosed in the ranking of 1 - 5 in brackets. Then the average ranking of each algorithm in the column and show the average ranking in the last row. Where the lower the average ranking, the better the algorithm. Furthermore, the best performance among the five comparing algorithms is demonstrated in bold font.

Table 23 The performance of multi-label algorithm in terms of Precision ↑.

Datasets	BP-MLL	ML-HARAM	ANN	LSIM	POL
Birds	0.151±0.048(3)	0.183±0.046(2)	0.067±0.022(5)	0.141±0.047(4)	0.220±0.058(1)
Enron	0.083±0.010(5)	0.450±0.012(4)	0.669±0.013(2)	0.677±0.029(1)	0.549±0.012(3)
Emotions	0.307±0.087(5)	0.373±0.060(3)	0.354±0.080(4)	0.656±0.044(2)	0.665±0.061(1)
Medical	0.030±0.008(5)	0.576±0.059(2)	0.063±0.052(4)	0.475±0.073(3)	0.765±0.055(1)
Yeast	0.383±0.043(5)	0.645±0.026(4)	0.709±0.020(2)	0.712±0.021(1)	0.658±0.024(3)
Scene	0.234±0.044(5)	0.322±0.228(2)	0.619±0.030(1)	0.254±0.189(4)	0.315±0.233(3)
Cal500	0.195±0.015(5)	0.501±0.120(3)	0.607±0.024(1)	0.572±0.018(2)	0.343±0.018(4)
Foodtruck	0.336±0.066(5)	0.691±0.062(2)	0.603±0.062(4)	0.664±0.022(3)	0.699±0.035(1)
Avg.rank	4.75	3.13	2.88	2.50	**2.13**

Table 24 The performance of multi-label algorithm in terms of Recall ↑.

Datasets	BP-MLL	ML-HARAM	ANN	LSIM	POL
Birds	0.090±0.019(3)	0.116±0.009(2)	0.072±0.035(5)	0.085±0.028(4)	0.202±0.051(1)
Enron	0.636±0.058(1)	0.466±0.030(4)	0.465±0.019(5)	0.478±0.011(3)	0.497±0.013(2)
Emotions	0.451±0.273(4)	0.461±0.118(3)	0.241±0.063(5)	0.681±0.069(2)	0.704±0.076(1)
Medical	0.522±0.135(3)	0.620±0.044(2)	0.052±0.045(5)	0.435±0.061(4)	0.715±0.042(1)
Yeast	0.585±0.101(4)	0.582±0.025(5)	0.585±0.017(3)	0.608±0.012(2)	0.622±0.020(1)
Scene	0.246±0.195(5)	0.382±0.269(3)	0.622±0.032(1)	0.545±0.181(2)	0.296±0.230(4)
Cal500	0.655±0.055(1)	0.232±0.072(4)	0.217±0.014(5)	0.251±0.005(3)	0.343±0.016(2)
Foodtruck	0.723±0.102(1)	0.486±0.034(2)	0.448±0.049(5)	0.485±0.044(3)	0.453±0.048(4)
Avg.rank	2.75	3.13	4.25	2.88	**2.00**

Table 25 The performance of multi-label algorithm in terms of F1 ↑.

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	2.88	3.25	2.63	**2.00**

Table 26 The performance of multi-label algorithm in terms of Macro Precision ↑.

Datasets	BP-MLL	ML-HARAM	ANN	LSIM	POL
Birds	0.179±0.027(3)	0.092±0.026(4)	0.070±0.030(5)	0.215±0.068(2)	0.471±0.058(1)
Enron	0.063±0.003(5)	0.130±0.016(4)	0.168±0.018(3)	0.210±0.014(1)	0.208±0.010(2)
Emotions	0.183±0.088(5)	0.293±0.055(4)	0.524±0.124(3)	0.668±0.056(1)	0.655±0.078(2)
Medical	0.030±0.009(4)	0.127±0.024(3)	0.017±0.007(5)	0.148±0.030(2)	0.236±0.018(1)
Yeast	0.297±0.026(5)	0.431±0.016(3)	0.414±0.039(4)	0.570±0.069(1)	0.499±0.024(2)
Scene	0.276±0.075(2)	0.253±0.055(3)	0.788±0.029(1)	0.251±0.134(4)	0.225±0.117(5)
Cal500	0.108±0.009(4)	0.116±0.012(3)	0.061±0.013(5)	0.138±0.011(2)	0.161±0.013(1)
Foodtruck	0.153±0.038(5)	0.174±0.046(4)	0.175±0.049(3)	0.268±0.048(2)	0.287±0.078(1)
Avg.rank	4.13	3.50	3.63	**1.88**	**1.88**

Table 27 The performance of multi-label algorithm in terms of Macro Recall ↑.

Datasets	BP-MLL	ML-HARAM	ANN	LSIM	POL
Birds	0.079+0.020(5)	0.088+0.035(4)	0.091+0.045(3)	0.094+0.039(2)	0.356+0.042(1)
Enron	0.447+0.044(1)	0.091+0.008(5)	0.096+0.006(4)	0.116+0.010(3)	0.139+0.006(2)
Emotions	0.442+0.271(3)	0.390+0.073(4)	0.221+0.055(5)	0.656+0.063(2)	0.692+0.074(1)
Medical	0.280+0.060(1)	0.158+0.025(3)	0.005+0.004(5)	0.094+0.026(4)	0.237+0.027(2)
Yeast	0.467+0.088(1)	0.400+0.016(2)	0.335+0.008(5)	0.343+0.015(4)	0.395+0.013(3)
Scene	0.547+0.182(2)	0.327+0.084(3)	0.634+0.031(1)	0.205+0.076(5)	0.217+0.075(4)
Cal500	0.483+0.033(1)	0.103+0.032(3)	0.052+0.004(5)	0.075+0.002(4)	0.160+0.013(2)
Foodtruck	0.403+0.127(1)	0.133+0.022(3)	0.118+0.016(5)	0.139+0.020(2)	0.124+0.014(4)
Avg.rank	**1.88**	3.38	4.13	3.25	2.38

Table 28 The performance of multi-label algorithm in terms of Macro F1 ↑.

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	3.38	3.88	2.75	**1.75**

Table 29 The performance of multi-label algorithm in terms of Micro Precision ↑.

Datasets	BP-MLL	ML-HARAM	ANN	LSIM	POL
Birds	0.778±0.076(2)	0.200±0.039(4)	0.161±0.052(5)	0.826±0.114(1)	0.586±0.049(3)
Enron	0.083±0.010(5)	0.421±0.019(4)	0.716±0.013(1)	0.683±0.020(2)	0.555±0.013(3)
Emotions	0.331±0.093(5)	0.360±0.041(4)	0.549±0.089(3)	0.679±0.040(1)	0.662±0.054(2)
Medical	0.030±0.008(5)	0.492±0.049(4)	0.766±0.329(3)	0.876±0.049(1)	0.772±0.054(2)
Yeast	0.378±0.038(5)	0.614±0.023(4)	0.712±0.018(2)	0.718±0.017(1)	0.666±0.021(3)
Scene	0.236±0.043(5)	0.288±0.196(4)	0.778±0.034(1)	0.373±0.262(2)	0.316±0.233(3)
Cal500	0.195±0.015(5)	0.343±0.009(3)	0.598±0.029(1)	0.562±0.016(2)	0.339±0.016(4)
Foodtruck	0.333±0.064(5)	0.646±0.054(3)	0.594±0.047(4)	0.669±0.044(2)	0.682±0.033(1)
Avg.rank	4.63	3.75	2.50	**1.50**	2.63

Table 30 The performance of multi-label algorithm in terms of Micro Recall ↑.

Datasets	BP-MLL	ML-HARAM	ANN	LSIM	POL
Birds	0.168+0.029(4)	0.220+0.019(2)	0.152+0.074(5)	0.173+0.050(3)	0.406+0.053(1)
Enron	0.615+0.062(1)	0.412+0.036(5)	0.439+0.021(4)	0.458+0.011(2)	0.453+0.015(3)
Emotions	0.453+0.270(3)	0.433+0.096(4)	0.245+0.054(5)	0.678+0.065(2)	0.704+0.076(1)
Medical	0.524+0.140(3)	0.598+0.041(2)	0.051+0.044(5)	0.440+0.058(4)	0.685+0.039(1)
Yeast	0.588+0.103(3)	0.577+0.021(5)	0.577+0.015(4)	0.606+0.010(2)	0.617+0.017(1)
Scene	0.544+0.182(2)	0.251+0.193(5)	0.612+0.033(1)	0.390+0.257(3)	0.299+0.221(4)
Cal500	0.653+0.054(1)	0.226+0.073(4)	0.209+0.014(5)	0.245+0.003(3)	0.339+0.018(2)
Foodtruck	0.352+0.028(5)	0.396+0.037(2)	0.354+0.037(4)	0.390+0.034(3)	0.663+0.115(1)
Avg.rank	2.75	3.63	4.13	2.75	**1.75**

Table 31 The performance of multi-label algorithm in terms of Micro F1 ↑.

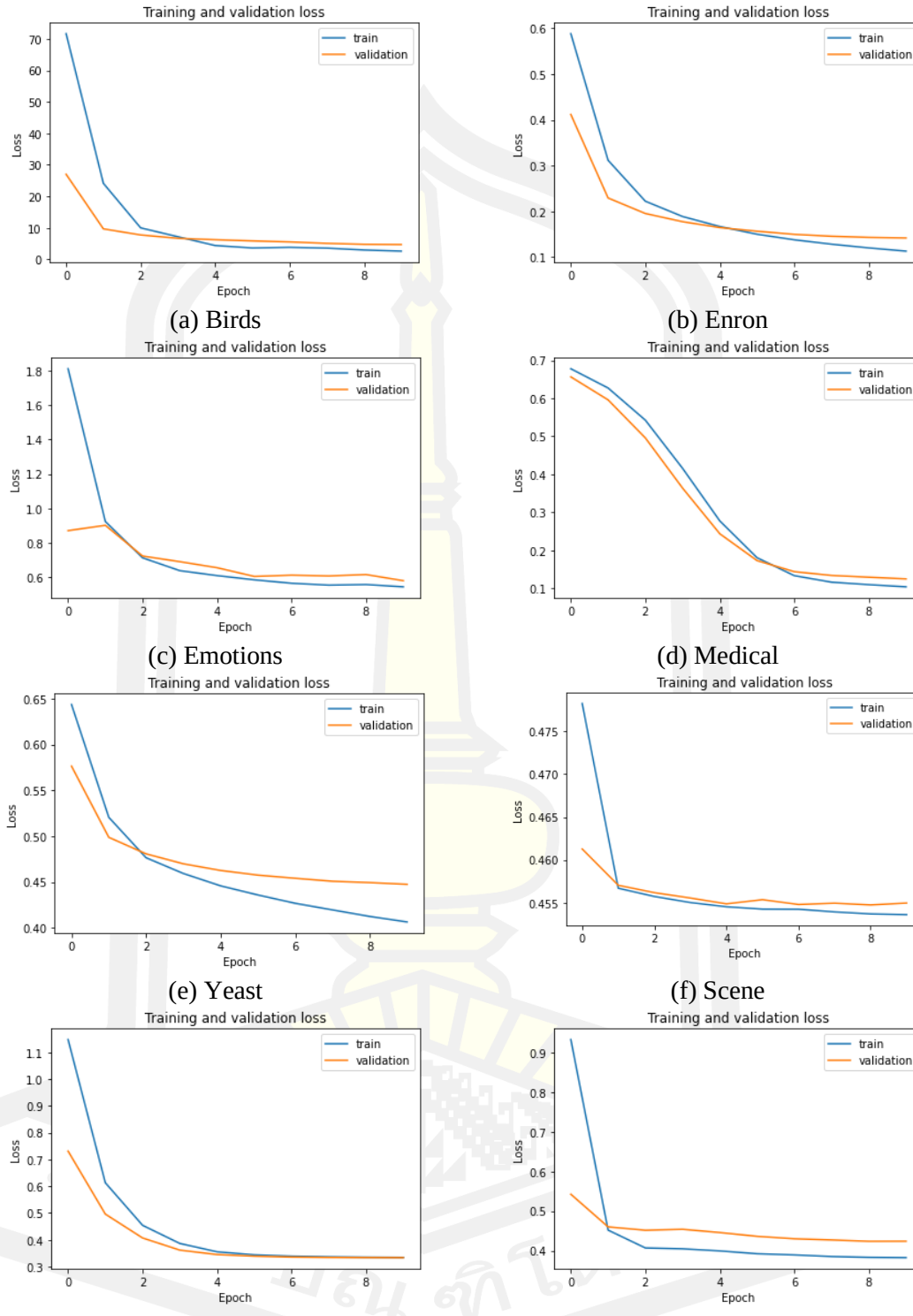
Datasets	BP-MLL	ML-HARAM	ANN	LSIM	POL
Birds	0.276±0.041(3)	0.207±0.022(4)	0.153±0.061(5)	0.281±0.070(2)	0.479±0.051(1)
Enron	0.146±0.016(5)	0.416±0.024(4)	0.544±0.017(2)	0.548±0.013(1)	0.499±0.013(3)
Emotions	0.336±0.109(4)	0.390±0.053(3)	0.333±0.054(5)	0.676±0.038(2)	0.680±0.056(1)
Medical	0.056±0.016(3)	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	4.00	3.50	3.50	**2.00**	**2.00**

Table 32 The performance of multi-label algorithm in terms of Hamming Loss ↓.

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	3.88	2.38	**1.50**	2.50

Tables 23 - 32 show the results of the experiment on the eight datasets with different methods. Each of the tables represents a different evaluation technique used in this study. For each table, the classification ranks are demonstrated. The smaller rank the better the technique. It can be observed that the proposed method provides promising results, resulting in low ranks for almost all the metrics. Considering Macro Recall, the proposed technique outputs marginally poor results compared to the BP-MLL.

To visualize the training behavior of the network, the graphical representation of the loss values and the validation are illustrated in Figure 18.



(a) Birds (b) Enron

(c) Emotions (d) Medical

(e) Yeast (f) Scene

(g) Cal500 (h) Foodtruck

Figure 18 Illustration of the visualization of the loss function of the datasets.

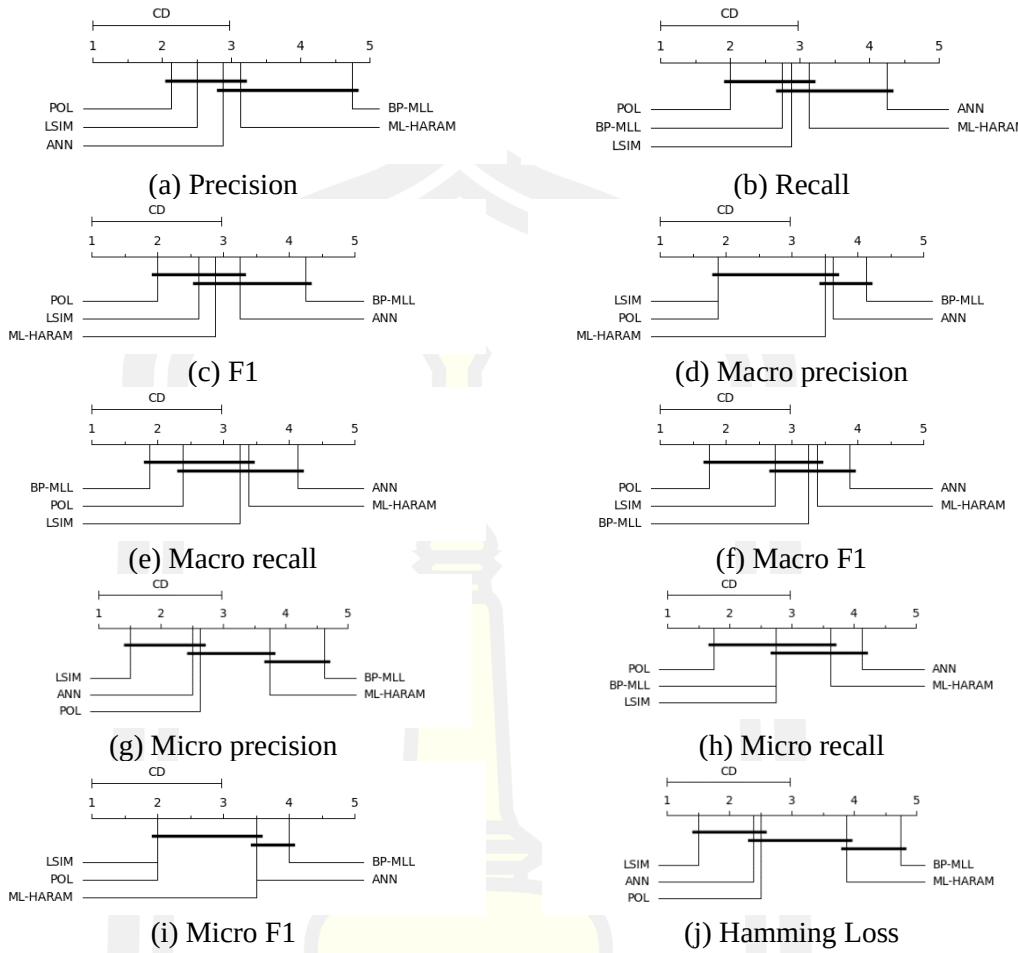


Figure 19 Comparison of proposed against other methods using Bonferroni-Dunn statistic (CD=1.974, α = 0.05)

Finally, we are interested in investigating the comparative significance of the MCL and the proposed technique in the experiments. Therefore, the Bonferroni-Dunn test (Demšar, 2006) is employed to serve the above purpose by treating it as the control method. Here, the difference between the average ranks of the proposed and one comparing algorithm is compared with the following Critical Difference (CD):

$$CD = q_{\propto} \sqrt{\frac{k(k+1)}{6N}} \quad (26)$$

For Bonferroni-Dunn set $q_{\propto}$ = 2.498 (Demšar, 2006) at significance level α = 0.05 and thus CD = 1.974 (k = 5, N=8). Accordingly, the performance between the proposed and one comparing method is deemed to be significantly different if their average ranks overall datasets differ by at least one CD.

Figure 19. shows the critical distance diagrams for each evaluation metric. 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. In each sub-figure, groups of algorithms that are not statistically different (their average rank is within one CD) from one another are connected.

4.5 Discussion

The objective of this work is to perform classification on multi-label data efficiently. We investigated the Neural Network-based architecture to perform the tasks. Soft-loss was introduced to assist the training to obtain a generalized model. This Soft-loss essentially interpolates the optimal network parameters (w) to converge toward label patterns exhibited in the dataset. The previous section demonstrates the results obtained from the experiments. We conducted the experiments using different ANN-based techniques, including the proposed methods. In addition, different data topologies were exploited to examine the methods' robustness.

BP-MLL demonstrates the promises in classifying the data. The technique does not provide promising results compared to ANN, ML-HARAM, and the proposed method. One of the key inspections is that BP-MLL essentially works well with the dataset that contains a high density of labels. The density of active labels in the multi-label configuration can help the model extract the embedded relationships of the label with respect to the given features. However, BP-MLL produces poor results on a general dataset with a different data topology. This can be noted that the technique may not be applicable to apply to the data with label sparsity.

The proposed method gives promising results from experiments compared to ANN and ML-HARAM. Compared to the BP-MLL method, it produced good results only with high-density labeled datasets. The proposed method produces excellent

results with distributed-labeled datasets. We can observe that our proposed method is significantly superior to the benchmark technique. The majority rank of the proposed technique with respect to different matrices is higher than the state-of-the-art method (>1.974).

4.6 Conclusion

This work presents a technique for classifying multi-label data instances. The work investigates ANN techniques to perform the classification. In general, feed-forward networks are applicable for classification problems. During a training process, the networks optimize the loss concerning the relation between predicted outcomes and the actual values. The work examines an alternative loss function approach used during training processes. In this work, the patterns of the multi-label class are explored. These patterns can essentially be used to construct a trained network that encourages the training to converge toward the existing patterns in the train data. Therefore, a weighted loss is proposed in this work. The loss is decomposed by two loss terms, i.e., (i) cross-entropy loss and (ii) pattern loss.

In the experimental results, the proposed method gave good results compared to the three methods, and the problem of sparse labeled datasets was solved with this approach. As a result, the results cover problems of high-density labeled datasets and sparse labeled datasets.

Chapter 5

Discussion

This thesis aims to present a method for improving the efficiency of multi-label classification (MLC) using the label correlation method based on the artificial neural network approach. The experiments were divided according to the research questions, which were set into three tasks. Firstly, the task was to compare the efficacy of the state-of-the-art MLC method with the chronic non-communicable disease dataset collected from Suthavej Hospital. The task was to examine the performance of different classification methods with respect to the specific data. Secondly, this thesis proposed and demonstrated a method to improve the efficiency of MLC essentially. The reconstruction feature method applying an AutoEncoder was introduced. The AutoEncoder encoded the relationship between data features and their labels. This resulted in a new features subset with smaller feature dimensions (compact features). Then, the generated features were used with several MLC approaches. Moreover, the work measured the efficiency of classification between native features with the proposed feature. Finally, this work introduced a method to improve the efficiency of MLC by investigating the pattern of class labels in the data applied with artificial neural network approaches. The work integrated the pattern information through the loss function in the model during the training process. The proposed method presented in this thesis can improve the performance of the classification.

This chapter will summarize and describe the findings from the proposed method. Then, the discussion of the results to answer the research questions in the description in the next section.

5.1. Answers to the Research Questions

In this section, the findings of the research findings will be discussed according to the research questions that have been laid out in Chapter 1.

RQ1: The MLC methods have been proposed to solve the problem of classifying more than one class, also known as multi-label, for more efficient classification. The initial or traditional method is presented in several groups as AM, PTM, and EM. These methods are referenced and are the base for developing new methods as BR, CC, LP, and ML-*K*NN. They provide excellent performance in multi-label classification. Therefore, this research is interested in applying the popular traditional method of MLC. It uses the Non-Communicable Disease (NCDs) diagnosis dataset to experiment and collect data from Suthavej Hospital. It is information on patients with chronic non-communicable diseases. For example, patients with diabetes often have hypertension. From the information, the patient data had more than one concomitant diagnosis. The MLC approach is needed to classify multiple diseases together for the above problem. The dataset is introduced into the MLC process using the defined methods. The results were compared to measure the efficiency of the classification of each method. Furthermore, the conclusion is which method can most accurately classify data for diagnosing chronic non-communicable diseases.

This research collected NCDs patient data from the hospital, which was used in the experiments. The data preparation process was carried out to obtain a dataset with multiple diseases concomitantly. Then, it focused on four diseases, namely diabetes, hypertension, cardiovascular, and stroke. The experiments were conducted with the NCDs dataset, then the performance of the results obtained from the MLC techniques was compared. The results and benchmarks showed that the RAkEL (Tsoumakas & Vlahavas, 2007b) method provided the highest accuracy compared with BR (Tsoumakas & Katakis, 2007b), CC (Read et al., 2011b), and ML-*K*NN (M. L. Zhang & Zhou, 2007) methods. The RAkEL method is considered to be a method applying an ensemble-based mechanism. The technique builds a random subset of the original labels to learn a single-label classifier (binary) to predict each element in the powerset of the subset. Therefore, ensemble-based classification methods can classify NCDs data better than other methods.

RQ2: The MLC performance improvements used features jointly with labels are well-known and allow for higher classification efficiency (Fan et al., 2021; J. Li et al., 2022; Nazmi et al., 2021). This work proposes a method for reconstructing features from learning the relationship between features and labels because features are an essential factor in classifying data in which labels use the AutoEncoder algorithm to learn to correlate features with labels. It will get a new feature that has changed the dimensions of the data may be increased or reduced. Nevertheless, the relationship of the feature data is more indicative of the label. This method permits the MLC algorithm to classify more accurately.

This research proposed a technique to improve the performance of MLC performance with a feature reconstruction method. The proposed feature reconstruction applied the AutoEncoder technique that intentionally encodes the input data instance to generate a compact feature representation of them. This work implemented two of the construction procedures. AutoEncoder alone (EN) was built to encode the feature subsets of the data instances. AutoEncoder with Target class (TEN) was constructed to derive a compact set of the data instances and maintain the contextual insights of the dataset, conveying the class-label representation to evaluate the performance of the proposed method 8-standard datasets were collected, derived from different domains, and different data settings. The experiments were conducted by applying six classifiers based on three different MLC techniques (i.e., PTM, AM, and EM). The experiments were separated into two folds. The first experiment explored the effectiveness of the TEN and EN in the feature reconstruction process. In comparison, the second experiment was objected to measuring the proposed technique's performance (TEN) compared with the original data feature used in MLC. The experimental results deliberately delineate the performance of the proposed technique. For all data sets, TEN essentially provides promising results, which is better than EN. The TEN works well with the Yeast (André Elisseeff, 2001) and Emotion (Trohidis et al., 2008) datasets, giving better results for all the MLC algorithms and the measurement metrics. The Yeast and Emotion are the only two datasets with high density (Tsoumakas, Spyromitros-Xioufis, et al., 2011). The density of the dataset in MLC indicates the well-presentation of the class labels. Therefore, TEN trends work well with the high-density dataset (well-presented data)

for MLC problems. In addition, the results obtained from the second experiment on the Yeast dataset show that the reconstruction technique is superior to the native data features (without feature transformation processes). In general, feature reconstruction can produce different sizes of compact features. Therefore, this work varied the sizes of the reconstructed features to observe the sensitivity of the technique. The results indicate that TEN gives better results than the native features for all MLC problems and measurement metrics.

RQ3: One of the possible solutions for performing the MLC task is to investigate the patterns of class labels in the dataset. The classification can be carried out in multi-class classification family schemes. Power subset is a general technique that converts MLC to multi-class problems. Based on the same principles, this research question will investigate the drive into using the pattern of a label (or data classes) in the data to assist the classification. The patterns of labels are used in training a model to obtain a generalized model for MLC.

One of the objectives of this work is to improve the performance of the classification of multi-label data. Therefore, this research investigated the Neural Network-based architecture to perform the tasks and answer the RQ3. Soft-loss was introduced to assist the training to obtain a generalized model. This Soft-loss essentially interpolates the optimal network parameters to converge toward label patterns exhibited in the dataset. This work conducted the experiments using different Artificial Neural Network (ANN) techniques. In addition, different data topologies were exploited to examine the robustness of the methods. The BP-MLL method (M. L. Zhang & Zhou, 2006) demonstrates the promises in classifying the data. The technique does not provide promising results compared to ANN, ML-HARAM, and the proposed method. One of the key inspections is that BP-MLL essentially works well with the dataset that contains a high density of labels. The density of active labels in the multi-label configuration can help the model extract the embedded relationships of the label with respect to the given features. However, BP-MLL produces poor results on a general dataset with a different data topology. This can be noted that the technique may not be applicable to apply to the data with label sparsity. In the experimental results, the proposed method gave good results compared to the three methods, and the problem of sparse labeled datasets was solved with this approach.

As a result, the results cover problems of high-density labeled datasets and sparse labeled datasets.

5.2. Future Work

This research concentrates on improving the efficiency of multi-label classification. The work developed two approaches for improving classification efficiency. First, feature engineering reconstructed features were introduced to capture the correlation between features and their labels, resulting in higher classification accuracy and rankings. Secondly, the work examined the feature correlation patterns with multiple label datasets resulting in a label correlation pattern. A model was introduced into the learning process with a customized loss function (in Artificial Neural Networks) that enforced the loss to consider the patterns of labels in the data. In future works, this research will focus on analyzing feature correlation with labels of multiple label datasets. Then, it will investigate appropriate techniques that incorporate the information of the label patterns for multi-label classification. There are a number of good algorithms that can be applied in the future. For example, popular deep learning algorithms like Convolutional Neural Network (CNN) algorithm works well with the image dataset (Anh et al., 2022; Bi et al., 2020; Noppitak & Surinta, 2021; Ou et al., 2022), and the Recurrent Neural Network (RNN) algorithm works well with time-series datasets (Wang et al., 2022; Zhao et al., 2018) . These algorithms can highly provide accurate classification. It may be applied to multi-label classification. Additionally, the researcher is interested in a multi-label dataset that is a real-world problem to propose a solution to the problem of MLC to solve problems in real-world applications.

REFERENCES

Alazaidah, R., & Ahmad, F. K. (2016). Trending Challenges in Multi Label Classification. *International Journal of Advanced Computer Science and Applications*, *7*. https://doi.org/10.14569/IJACSA.2016.071017

Alluwaici, M., Junoh, A. K., & Alazaidah, R. (2020). New Problem Transformation Method Based on the Local Positive Pairwise Dependencies Among Labels. *Journal of Information \& Knowledge Management*, *19*(01), 2040017.

André Elisseeff, J. W. (2001). A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems 14*. https://papers.nips.cc/paper/2001/hash/39dcaf7a053dc372fbc391d4e6b5d693-Abstract.html

Anh, P. T. Q., Thuyet, D. Q., & Kobayashi, Y. (2022). Image classification of root-trimmed garlic using multi-label and multi-class classification with deep convolutional neural network. *Postharvest Biology and Technology*, *190*, 111956. https://doi.org/10.1016/J.POSTHARVBIO.2022.111956

Benites, F., & Sapozhnikova, E. (2016). HARAM: A Hierarchical ARAM Neural Network for Large-Scale Text Classification. *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, 847–854. https://doi.org/10.1109/ICDMW.2015.14

Benites, F., & Sapozhnikova, E. (2017). Improving scalability of ART neural networks. *Neurocomputing*, *230*, 219–229. https://doi.org/10.1016/J.NEUCOM.2016.12.022

Bernardini, F. C., Silva, R. B. da, Rodovalho, R. M., & Meza, E. B. M. (2014). Cardinality and Density Measures and Their Influence to Multi-Label Learning Methods. *Learning and Nonlinear Models*, *12*(1), 53–71. https://doi.org/10.21528/LNLM-VOL12-NO1-ART4

Bi, L., Feng, D. D., Fulham, M., & Kim, J. (2020). Multi-Label classification of multi-modality skin lesion via hyper-connected convolutional neural network. *Pattern Recognition*, *107*, 107502. https://doi.org/10.1016/J.PATCOG.2020.107502

Bogatinovski, J., Todorovski, L., D\vzeroski, S., & Kocev, D. (2021). Comprehensive Comparative Study of Multi-Label Classification Methods. *ArXiv*, *abs/2102.0*.

Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, *37*(9), 1757–1771.

https://doi.org/10.1016/j.patcog.2004.03.009

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chandran, S. A., & Panicker, J. R. (2017). An efficient multi-label classification system using ensemble of classifiers. *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 1133–1136. https://doi.org/10.1109/ICICICT1.2017.8342729

Chen, S.-F., Chen, Y.-C., Yeh, C.-K., & Wang, Y.-C. F. (2017). *Order-Free RNN with Visual Attention for Multi-Label Classification*. http://arxiv.org/abs/1707.05495

Chen, W.-J., Shao, Y.-H., Li, C.-N., & Deng, N.-Y. (2016). MLTSVM: a novel twin support vector machine to multi-label learning. *Pattern Recognition*, *52*, 61–74.

Cheng, D., Zhang, S., Deng, Z., Zhu, Y., & Zong, M. (2014). *kNN Algorithm with Data-Driven k Value*. 499–512. https://doi.org/10.1007/978-3-319-14717-8_39

Cheng, Y., Zhao, D., Wang, Y., & Pei, G. (2019). Multi-label learning with kernel extreme learning machine autoencoder. *Knowledge-Based Systems*, *178*, 1–10. https://doi.org/10.1016/j.knosys.2019.04.002

Cherman, E., Monard, M.-C., & Metz, J. (2011). Multi-label Problem Transformation Methods: a Case Study. *CLEI Electron. J.*, *14*. https://doi.org/10.19153/cleiej.14.1.4

Clare, A., & King, R. D. (2001). Knowledge Discovery in Multi-label Phenotype Data. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2168*, 42–53. https://doi.org/10.1007/3-540-44794-6_4

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, *7*, 1–30. https://doi.org/10.5555/1248547.1248548

Deng, Z., Wang, S., & Chung, F. (2013a). A minimax probabilistic approach to feature transformation for multi-class data. *Applied Soft Computing*, *13*(1), 116–127. https://doi.org/https://doi.org/10.1016/j.asoc.2012.08.003

Deng, Z., Wang, S., & Chung, F. L. (2013b). A minimax probabilistic approach to feature transformation for multi-class data. *Applied Soft Computing*, *13*(1), 116–127. https://doi.org/10.1016/J.ASOC.2012.08.003

Elisseeff, A., & Weston, J. (2001). Kernel methods for Multi-labelled classification and Categorical regression problems. *In Advances in Neural Information Processing Systems 14*, 681–687.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., & Dehmer, M. (2020). An Introductory Review of Deep Learning for Prediction Models With Big Data. *Frontiers in Artificial Intelligence*, *3*. https://doi.org/10.3389/frai.2020.00004

Ezzine, I., & Benhlima, L. (2018). *A Study of Handling Missing Data Methods for Big Data*. 498–501. https://doi.org/10.1109/CIST.2018.8596389

Fan, Y., Liu, J., Weng, W., Chen, B., Chen, Y., & Wu, S. (2021). Multi-label feature selection with local discriminant model and label correlations. *Neurocomputing*, *442*, 98–115. https://doi.org/10.1016/J.NEUCOM.2021.02.005

Gao, W., Hu, J., Li, Y., & Zhang, P. (2020). Feature Redundancy Based on Interaction Information for Multi-Label Feature Selection. *IEEE Access*, *8*, 146050–146064. https://doi.org/10.1109/ACCESS.2020.3015755

Gibaja, E., Moyano, J., & Ventura, S. (2016). An ensemble-based approach for multi-view multi-label classification. *Progress in Artificial Intelligence*, *5*. https://doi.org/10.1007/s13748-016-0098-9

Godbole, S., & Sarawagi, S. (2004). Discriminative Methods for Multi-labeled Classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3056*, 22–30. https://doi.org/10.1007/978-3-540-24775-3_5

Guozhu, D., & Huan, L. (Eds.). (2018). *Feature Engineering for Machine Learning and Data Analytics - 1st Edit*. CRC Press. https://www.routledge.com/Feature-Engineering-for-Machine-Learning-and-Data-Analytics/Dong-Liu/p/book/9780367571856

Hafeez, G., Khan, I., Jan, S., Shah, I. A., Khan, F. A., & Derhab, A. (2021). A novel hybrid load forecasting framework with intelligent feature engineering and optimization algorithm in smart grid. *Applied Energy*, *299*, 117178. https://doi.org/https://doi.org/10.1016/j.apenergy.2021.117178

Herrera, F., Charte, F., Rivera, A. J., & Del Jesus, M. J. (2016). Multilabel classification: Problem analysis, metrics and techniques. In *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Springer. https://doi.org/10.1007/978-3-319-41111-8

Herrera, F., Charte, F., Rivera, A. J., del Jesus, M. J., Herrera, F., Charte, F., Rivera, A. J., & del Jesus, M. J. (2016). Multilabel Classification. In *Multilabel*

Classification (pp. 1–16). Springer International Publishing. https://doi.org/10.1007/978-3-319-41111-8_1

Huang, J., Li, G., & Wu, X. (2018). Joint Feature Selection and Classification for Multilabel Learning. *IEEE Transactions on Cybernetics*, *PP*, 1–14. https://doi.org/10.1109/TCYB.2017.2663838

International Health Policy Program. (2015). *Disability-Adjusted Life Year :DALY*. The Graphico Systems. http://www.thaincd.com/document/file/download/knowledge/report_BOD_2556.pdf

Jin, W., Hong, W., Cuiping, X., Weihua, O., Qiaosong, C., & Xin, D. (2017). Ensembles of classifier chains for multi-label classification based on Spark. *Journal of University of Science and Technology of China*, *47*(4), 350. https://doi.org/10.3969/J.ISSN.0253-2778.2017.04.010

Kimura, K., Kudo, M., Sun, L., & Koujaku, S. (2016). Fast random k-labELsets for large-scale multi-label classification. *2016 23rd International Conference on Pattern Recognition (ICPR)*, 438–443. https://doi.org/10.1109/ICPR.2016.7899673

Li, J., Li, P., Hu, X., & Yu, K. (2022). Learning common and label-specific features for multi-Label classification with correlation information. *Pattern Recognition*, *121*, 108259. https://doi.org/10.1016/J.PATCOG.2021.108259

Li, R., Liu, W., Lin, Y., Zhao, H., & Zhang, C. (2017). An Ensemble Multilabel Classification for Disease Risk Prediction. *Journal of Healthcare Engineering*, *2017*. https://doi.org/10.1155/2017/8051673

Lian, S. ming, Liu, J. wei, Lu, R. kun, & Luo, X. lin. (2019). Captured multi-label relations via joint deep supervised autoencoder. *Applied Soft Computing Journal*, *74*, 709–728. https://doi.org/10.1016/j.asoc.2018.10.035

Liou, C.-Y., Cheng, W.-C., Liou, J.-W., & Liou, D.-R. (2014). Autoencoder for words. *Neurocomputing*, *139*, 84–96.

Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. (2016, November 11). Learning to diagnose with LSTM recurrent neural networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. http://zacklipton.com

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012a). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, *45*(9), 3084–3104. https://doi.org/10.1016/j.patcog.2012.03.004

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012b). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, *45*(9), 3084–3104. https://doi.org/10.1016/j.patcog.2012.03.004

Mandziuk, J., & Zychowski, A. (2019). Dimensionality Reduction in Multilabel Classification with Neural Networks. *Proceedings of the International Joint Conference on Neural Networks*, *2019-July*. https://doi.org/10.1109/IJCNN.2019.8852156

Maxwell, A., Li, R., Yang, B., Weng, H., Ou, A., Hong, H., Zhou, Z., Gong, P., & Zhang, C. (2017). Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC Bioinformatics*, *18*(Suppl 14). https://doi.org/10.1186/s12859-017-1898-z

Mencía, E. L., Fürnkranz, J., Hüllermeier, E., & Rapp, M. (2018). *Learning Interpretable Rules for Multi-Label Classification*. 81–113. https://doi.org/10.1007/978-3-319-98131-4_4

Ministry of Public Health. (2017). *Standard structure of medical and health information from Ministry of Public Health, Thailand*. http://bps.moph.go.th/new_bps/sites/default/files/2.43file_Structure_2560-11-08_V2.3.pdf

Moral García, S., Mantas, C., Castellano, F., & Abellán, J. (2019). Ensemble of classifier chains and Credal C4.5 for solving multi-label classification. *Progress in Artificial Intelligence*, *8*. https://doi.org/10.1007/s13748-018-00171-x

Nápoles, G., Bello, M., & Salgueiro, Y. (2021). Long-term Cognitive Network-based architecture for multi-label classification. *Neural Networks*, *140*, 39–48. https://doi.org/10.1016/j.neunet.2021.03.001

Nazmi, S., Yan, X., Homaifar, A., & Anwar, M. (2021). Multi-label classification with local pairwise and high-order label correlations using graph partitioning. *Knowledge-Based Systems*, *233*, 107414. https://doi.org/10.1016/J.KNOSYS.2021.107414

Noppitak, S., & Surinta, O. (2021). *ICIC Express Letters ICIC International©2021 ISSN*. *15*(6), 531–543. https://doi.org/10.24507/icicel.15.06.531

Ou, X., Gao, L., Quan, X., Zhang, H., Yang, J., & Li, W. (2022). BFENet: A two-stream interaction CNN method for multi-label ophthalmic diseases classification with bilateral fundus images. *Computer Methods and Programs in Biomedicine*, *219*, 106739. https://doi.org/10.1016/J.CMPB.2022.106739

Patterson, J., & Gibson, A. (2017). *Deep Learning*. O'Reilly Media, Inc.

Prajapati, P., & Thakkar, A. (2021). Performance improvement of extreme multi-label classification using K-way tree construction with parallel clustering algorithm. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2021.02.014

Pushpa, M., & Karpagavalli, S. (2017). Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification. *Procedia Computer Science*, *115*, 572–579. https://doi.org/https://doi.org/10.1016/j.procs.2017.09.116

Read, J., Pfahringer, B., & Holmes, G. (2008). Multi-label classification using ensembles of pruned sets. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 995–1000. https://doi.org/10.1109/ICDM.2008.74

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5782 LNAI*(PART 2), 254–269. https://doi.org/10.1007/978-3-642-04174-7_17

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011a). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359. https://doi.org/10.1007/s10994-011-5256-5

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011b). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359. https://doi.org/10.1007/s10994-011-5256-5

Read, J., Puurula, A., & Bifet, A. (2014). Multi-label Classification with Meta-Labels. *2014 IEEE International Conference on Data Mining*, 941–946. https://doi.org/10.1109/ICDM.2014.38

Read, J., Reutemann, P., Pfahringer, B., & Holmes, G. (2016). Meka: A Multi-label/Multi-target Extension to Weka. *JMLR*, *17*, 1–5.

Rokach, L., Schclar, A., & Itach, E. (2013). Ensemble Methods for Multi-label Classification. *Expert Systems with Applications*, *41*. https://doi.org/10.1016/j.eswa.2014.06.015

Runzhi Li, Hongling Zhao, Yusong Lin, Maxwell, A., & Chaoyang Zhang. (2016). Multi-label classification for intelligent health risk prediction. *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 986–993. https://doi.org/10.1109/BIBM.2016.7822657

Sangkatip, W., & Phuboon-Ob, J. (2020). Non-Communicable Diseases Classification using Multi-Label Learning Techniques. *International Conference on*

Information Technology (InCIT), 17–21. https://doi.org/10.1109/InCIT50588.2020.9310978

Sousa, R., & Gama, J. (2016). *Online Multi-label Classification with Adaptive Model Rules*. *9868*, 58–67. https://doi.org/10.1007/978-3-319-44636-3_6

Szymański, P., & Kajdanowicz, T. (2017). A scikit-based Python environment for performing multi-label classification. *ArXiv E-Prints*.

Tanaka, E. A., Nozawa, S. R., Macedo, A. A., & Baranauskas, J. A. (2015). A multi-label approach using binary relevance and decision trees applied to functional genomics. *Journal of Biomedical Informatics*, *54*, 85–95. https://doi.org/10.1016/j.jbi.2014.12.011

Trohidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. (2008). ISMIR 2008-Session 3a-Content-Based Retrieval, Categorization and Similarity 1. *International Conference on Music Information Retrieval (ISMIR 2008)*, 325–330. http://www.musicovery.com/

Tsoumakas, G., & Katakis, I. (2007a). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13. https://doi.org/10.4018/jdwm.2007070101

Tsoumakas, G., & Katakis, I. (2007b). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13. https://doi.org/10.4018/jdwm.2007070101

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2008). *Effective and Efficient Multilabel Classification in Domains with Large Number of Labels*. 30–44.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2011). Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, *23*(7), 1079–1089. https://doi.org/10.1109/TKDE.2010.164

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., & Vlahavas, I. (2011). MULAN: A Java library for multi-label learning. *Journal of Machine Learning Research*, *12*, 2411–2414.

Tsoumakas, G., & Vlahavas, I. (2007a). Random k-labelsets: An ensemble method for multilabel classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4701 LNAI*, 406–417. https://doi.org/10.1007/978-3-540-74958-5_38

Tsoumakas, G., & Vlahavas, I. (2007b). Random k-Labelsets: An Ensemble Method for Multilabel Classification. In J. N. Kok, J. Koronacki, R. L. de Mantaras, S.

Matwin, D. Mladenič, & A. Skowron (Eds.), *Machine Learning: ECML 2007* (pp. 406–417). Springer Berlin Heidelberg.

Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., & Chervyakov, N. I. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, *177*, 232–243. https://doi.org/10.1016/j.matcom.2020.04.031

Wang, B., Hu, X., Zhang, C., Li, P., & Yu, P. S. (2022). Hierarchical GAN-Tree and Bi-Directional Capsules for multi-label image classification. *Knowledge-Based Systems*, *238*, 107882. https://doi.org/10.1016/J.KNOSYS.2021.107882

WHO. (2016). *ICD-10 Version:2016*. 2016. https://icd.who.int/browse10/2016/en

WHO. (2021). *Noncommunicable diseases*. 13 April 2021. https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases

Wu, X.-Z., & Zhou, Z.-H. (2017). A Unified View of Multi-Label Performance Measures. *ICML*.

Xiao, Y., Li, Y., Yuan, J., Guo, S., Xiao, Y., & Li, Z. (2021). History-based attention in Seq2Seq model for multi-label text classification. *Knowledge-Based Systems*, *224*, 107094. https://doi.org/10.1016/j.knosys.2021.107094

Yeh, C.-K., Wu, W.-C., Ko, W.-J., & Wang, Y.-C. F. (2017). *Learning Deep Latent Spaces for Multi-Label Classification*. http://arxiv.org/abs/1707.00418

Zhang, M.-L. (2011). Lift: Multi-Label Learning with Label-Specific Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*, 1609–1614. https://doi.org/10.1109/TPAMI.2014.2339815

Zhang, M. L., & Zhou, Z. H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, *18*(10), 1338–1351. https://doi.org/10.1109/TKDE.2006.162

Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048. https://doi.org/10.1016/j.patcog.2006.12.019

Zhang, M. L., & Zhou, Z. H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *26*(8), 1819–1837. https://doi.org/10.1109/TKDE.2013.39

Zhang, X., Zhao, H., Zhang, S., & Li, R. (2019). A Novel Deep Neural Network Model for Multi-Label Chronic Disease Prediction. *Frontiers in Genetics*, *10*(April). https://doi.org/10.3389/fgene.2019.00351

Zhao, B., Li, X., Lu, X., & Wang, Z. (2018). A CNN–RNN architecture for multi-label weather recognition. *Neurocomputing*, *322*, 47–57. https://doi.org/10.1016/J.NEUCOM.2018.09.048

REFERENCES

BIOGRAPHY

NAME	Worawith Sangkatip
DATE OF BIRTH	15 July 1987
PLACE OF BIRTH	Samut Prakan Province.
ADDRESS	368 Moo 10, Wang Nang Subdistrict, Mueang District, Maha Sarakham Province.
POSITION	Lecturer
PLACE OF WORK	Faculty of Information Technology, Rajabhat Maha Sarakham University.
EDUCATION	- 2022 Ph.D. Information Technology, Mahasarakham University. - 2013 M.Sc. Information Technology, Mahasarakham University. - 2010 B.Sc. Information Communication Technology, Mahasarakham University.
Research grants & awards	Graduate Publishing Promotion Scholarship from Mahasarakham University.
Research output	- Sangkatip, W., & Phuboon-Ob, J. (2020). Non-Communicable Diseases Classification using Multi-Label Learning Techniques. 2020 - 5th International Conference on Information Technology (InCIT), 17–21. https://doi.org/10.1109/InCIT50588.2020.9310978