Sudan University of Science and Technology
College of Graduate Studies

A New Hierarchical Support Vector Machine based Model for Classification of Imbalanced Multi-class Data

نموذج جديد مرتكز على آلة المتجهات الداعمة لتصنيف البيانات غير المتوازنة متعددة الأصناف

By:
Hanaa Sameeh A.Aziz Othman

Supervised by:
Dr. Mohamed ElHafiz Mustafa

March 2017

A New Hierarchical Support Vector Machine based Model for Classification of Imbalanced Multi-class Data

by

Hanaa Sameeh A.Aziz Othman

A dissertation submitted to College of Graduate Studies

In partial fulfillment of the requirements for the degree of

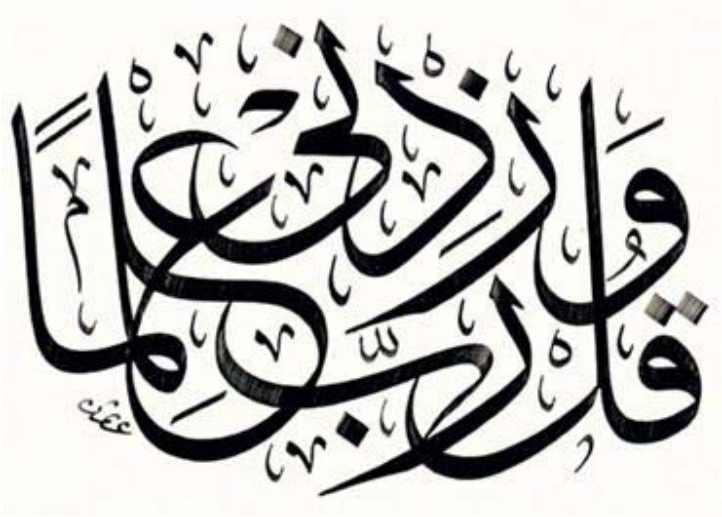
DOCTOR OF PHILOSOPHY

In Computer Science

Major Professor: Mohmmed ElHafiz Mustafa

Sudan University for Science &Technology

March 2017



سورة طه – الآية: ١١٤

DEDICATION

To my Parents, Husband, Kids, all my family members, My Teachers and to everyone encouraged and supported me.

Acknowledgement

First, this work wouldn't have been fulfilled unless ALLAH willed and granted its success and completion, so absolute endless thanks for him.

I wish to express my thanks to my supervisor Dr. Mohmmed Elhafiz for the valuable time he gave me, his encouragement and advice.

Thanks to the kind staff of SUST, particularly Professor Izzeldin Mohammed Othman, for his support and help.

I'd like to thank my parents for everything.

I am also grateful to my husband who supported me too much and my kids who were always patient and indulgent.

I would like to dedicate special thanks to my friends and colleagues who have listened to my questions, provided answers and advices, picked me out of very hard moments. Dr. Eiman Kambal and Dr. Limyaa Rahmat Allah in particular.

Great appreciation for all who helped me in a way or another.

Table of Contents:

Table of Tables

Table of Figures

ABSTRACT

The Imbalance Multi-class learning problem is one of the challenging problems in supervised machine learning. The imbalance nature of the data – which is owning skewed distribution of samples in different classes –as well as being multiclass – where an instance could be assigned to more than one class - lead to many vital problems in both learning and performance evaluation processes.

The research problem could be epitomized in finding more accurate classification results for such kind of data. So, its methodology is based on proposing new classification hierarchical method based on Multi-Class Support Vector Machine (Multi-Class SVM). The model rebalances the data via grouping small classes in bigger classes (artificial classes). Then it classifies the compound classes into its constituent classes at later stage. Experiments were applied on nine different Multiclass imbalanced datasets from U.C.I. repository.

The experiments show that the new hierarchical model enhances the classification results comparing with the classification results of some state-of-the-art solution, even when empowered with weight for minority instances, considering four different performance metrics. They also exhibit that the model is not only successful in treating the imbalance problem simply without computational efforts or algorithmic modification, but also it does not require any data pre-processing step as many other solutions need. So, there is no additional adaptation neither on the data level, nor on the algorithmic level. Moreover, the experiments showed that the model performs well even when the ratio between minority and majority samples is high. They also demonstrate that the model works better with large number of classes of a dataset and perform poorly with the dataset that owns little number of classes that could not be combined into artificial classes of nearly balanced numbers of examples.

مستخلص الدراسة:

إن مشكلة التعلم من البيانات الغير متوازنة – من ناحية عدد عينات فئاتها – متعددة الفئات – وهي التي تضم عينات يجب تصنيفها لواحدة من مجموعة من الفئات (أكثر من مجموعتين) – هي إحدى المهام المعقدة من مهام تعلم الآلة ذي الأسلوب المراقب، حيث أن طبيعتي البيانات تؤثران سلبا على أداء عدد من خوارزميات التعلم التقليدية وعلى خوارزميات تقييم أدائها.

تتلخص مشكلة هذه الدراسة في كيفية التوصل إلى تصنيف هذا النوع من البيانات بصورة دقيقة ، وبالتالي فقد إستندت منهجية الدراسة على تصميم نموذجا هرميا جديدا للتصنيف يرتكزعلى خوارزميتي آلة المتجهات الداعمة و آلة المتجهات الداعمة للفئات المتعددة، يقوم بإعادة التوازن للعينات عبر تجميع الفئات الصغيرة داخل فئات (افتراضية) أكبر، ومن ثم يجري عملية تصنيف الفئات المركبة الجديدة للفئات المكونة لها في مرحلة لاحقة.

تم تطبيق هذا النموذج على تسع مجموعات من مجموعات البيانات المختلفة المستوردة من مستودع U.C.I لمجموعات البيانات المخصصة للأبحاث.

أظهرت التجارب تحسن نتائج تصنيف النموذج المقترح بمقارنته مع خوارزميات تصنيف أُخرى اعتمدت من دراسات سابقة، حتى عند تزويد هذه الخوارزميات بأوزان لتصنيف العينات الأقل ، وذلك وفقا لأربعة معايير لتقييم أداء الخوارزميات. كما أوضحت التجارب أن النموذج المقترح لم ينجح فقط في التعامل مع مشكلة عدم توازن عدد العينات بسهولة وبدون أعباء معالجة اضافية، أو تعديلات في صميم الخوارزميات المستخدمة في النموذج، ولكنه أيضا لا يتطلب أيً نوع من المعالجة المسبقة للعينات كما تتطلب بعض التقنيات السابقة لتصنيف مثل هذه البيانات، وبالتالي لا تغييرات على

مستوى الخوارزميات ولا على مستوى العينات. إضافة إلى ما سبق، فالنموذج يظهر نتائجا جيدة حتى عندما يكون معدل عدم التوازن بين عينات مجموعات البيانات عاليا، وأثبتت التجارب أن النموذج المقترح له القدرة على العمل بشكل أفضل كلما كبرعدد فئات مجموعة البيانات، ويسوء أداؤه إذا قل عدد الفئات للحد الذي لا يمكن معه اعادة تجميعها في فئات إفتراضية متوازنة تقريبا.

1. CHAPTER ONE: INTRODUCTION

1.1. Overview

Learning from imbalanced datasets is a challengeable problem. It exists in a wide variety of real-world applications, where the class of interest is generally a small fraction of the total instances meanwhile misclassification of such instances is much expensive. The challenge becomes more complicated when the imbalanced data has also multiclass nature. While there is a significant focus on the class imbalance problem for binary class datasets, multi-class datasets haves received less attention. The Imbalanced Multiclass problem belongs to the supervised machine learning tasks, where each instance should be assigned to one of N different classes that have unequal sample sizes. It owns more complex characteristics that introduces more obstacles and issues to be considered during learning process and requires new understandings, principles, algorithms, and tools. Going a step further, many of the evaluation metrics deployed in practice vary significantly across the class imbalance and Multiclass data literatures, so far, no single measure can assess the performance of each learning machine and could be applied over such data because of the existence of undesirable properties and the implementation complexity.

The rest of this chapter is structured as follows: the subsection 1.2 addresses the research question, followed by subsection 1.3 in which problem goal and objectives are pointed out. The importance of this research is highlighted in subsection 1.4. The research contribution is clarified in subsection 1.5. Some terminologies that are used in this research are

illustrated in subsection 1.6. Finally, a general description of thesis organization is presented in section 1.7.

1.2. Research question:

Can we be able to provide more accurate classification results for the multiclass imbalanced data in such a way that is simple to be implemented, since the previous introduced solutions required some tuning either on data or on the utilized algorithms?

1.3. Research Goal and Objectives:

The main aim of this study is at getting more precise assignment of the few or the rare examples to their minority classes, and to enhance the predictability of the SVM classifier in the unseen data, hence we're looking for better overall performance when data is imbalanced and Multi-classed as well.

This aim could be accomplished by carrying out the following objectives:

- Review the different strategies for dealing with imbalanced data and those dedicated for Multiclass data as well concentrating on imbalanced multiclass datasets solutions and methods. Then address their pons and cons to develop a suitable method to refine the dataset flaws then get better performance of the classifier.
- Investigate the data format and characteristics and test its imbalance limits.
- Develop a model for the classification process basing on Support Vector Machine (SVM) and Multi-Class SVM to figure out how deploying multiclass SVM is effective to build a classification model that can classify the minority over majority instances in the presence of Multiclass imbalanced data accurately. Beside deciding

whether it is better to use it alone in one stage learning model, or in a hierarchical one?

- Investigate the overall performance through suitable assessment metrics empirically.

1.4. Research Significance:

The importance of this research rises from the fact that imbalanced multiclass data is produced from many real sensitive applications and fields in our life, such as the medical diagnosis, fraud detection, risk management in telecommunications, intrusion detection...etc. Another fact is that most efforts that have been proposed for solving the Multiclass issue so far are focused on two-class imbalance problems, meanwhile learning from multiclass imbalanced is more challenging and problematic and need more and more investigations.

1.5. Research Contribution:

- This study contributes pointing out some unsolved questions and complications caused by imbalanced multiclass data, and examines the generalization ability of strong classic pattern recognition tool, that have been widely used (Support Vector Machine).
- It presents a novel hierarchical model based on SVM and MultiSVM.
- It introduces a new Grouping algorithm for the dataset classes that don't depend on the similarities between instances such as the way the clustering technique works, instead, it originates new balanced artificial groups from the original imbalanced classes. So, this model does not use any fixed hierarchy based on features and/or classes.
- The new model gets the benefit of the black box of the nature of the Support Vector Machine to group the heterogeneous different classes.
- The classification process is divided into levels in such a way that does not utilizes any fixed hierarchy based on features and/or classes.

- By being based on this Grouping algorithm, it provides no computational complexity or algorithmic modification or even data distribution adjustment, so, it is different from common hierarchical methods which use supervised learning.
- The new model performs well even when increasing the number of classes.
- Results show that the proposed method is more successful than utilizing a Support Vector Machine even with support of using weight during the classification process. It performs well when the class imbalanced ratio (the number of minority class samples over majority class samples) is not extremely high.

1.6. Research Terminologies

- **Learning from Imbalanced Data**

It is learning process from a dataset that exhibits a significant unequal distribution of examples between its classes or within a single class.

- **Learning from Multiclass Data**

It is a learning process where each data point belongs to one of ***N*** different classes, so its aim is at constructing a function that will correctly assign each new data point to one of ***N*** classes that it is belongs to.

- **Learning from Multiclass Imbalanced Data**

It is a learning task from data where each instance should be assigned to one of ***N*** different classes that suffer unequal samples sizes.

- **SVM (Support Vector Machine):**

It is a learning algorithm that tries to find the optimal separating hyperplane that effectively separates these data points into two classes. It can provide relatively robust classification results when applied to

imbalanced data sets. While **Multiclass SVM** is the extended modified version deals with Multiclass data.

1.7. Research Outline

Chapter one includes a Background, Problem definition, Research goal and objectives, and Research importance and contribution. While Chapter two contains the literature review and the related work that discusses: Imbalanced data, Multiclass data, class Imbalance learning methods for Support Vector Machines, Chapter three highlights the research methodology and implementation. Chapter four points out the performance evaluation process and its different metrics. Chapter five discusses the research Results, and Chapter six indicates the conclusions, recommendations and future work**.**

2. CHAPTER TWO: LITERATURE REVIEW AND RELATED WORK

2.1. Introduction

This chapter focuses on providing a critical analysis of the problem nature, the state-of-the-art approaches, of the Imbalanced, Multi-class and Multiclass Imbalanced learning algorithms. Furthermore, it highlights the major opportunities, challenges and solutions introduced in this field.

2.2. Imbalance Data

2.2.1. What is an Imbalanced Dataset?

It's a data set that exhibits a significant unequal distribution of examples between its classes or within a single class [1].

2.2.2. What is Imbalance Learning?

(Haibo He) had defined the imbalance learning as follows: "The learning process for data representation and information extraction with severe data distribution skews to develop effective decision boundaries to support the decision-making process". The learning process could involve supervised learning, unsupervised learning, semi-supervised learning, or a combination [1]. Classification of Imbalanced data in general refers to assignment of the skewed distributed instances to one of two possible classes, which is called – in more accurate words - Binary Imbalanced Classification or Two classes scenario.

2.2.3. What are the problems of learning from imbalanced data?

There many obstacles appear when dealing with imbalanced data: firstly, standard classification algorithms are often biased towards the majority class, because they target decreasing global quantities such as the error rate, increasing the classifier accuracy regardless the data distribution, so examples from the majority class are classified accurately, while the minority examples is probably misclassified or overlooked. Secondly, the induction rules of the minority examples are weak, since they depend on finding the similarities between the examples which are less represented considering the minority ones. Thirdly, when deploying the learning algorithms that are based on greedy search algorithms and/or divide and conquer approach such as decision trees, the imbalanced data sets exploit insufficiencies in electing the best feature as the splitting criterion (e.g., information gain) at each node of the decision tree. Also, the more partitioning the instance space (and the examples that belong to these spaces) into smaller and smaller pieces, the more obtaining fewer leaves that describe minority examples from which the rare patterns must be identified. In other words, the more fragment data, the more get fewer existence of minority class examples. This issue is related to the problems of relative and absolute imbalances. So, in both cases, the effects of imbalanced data on decision tree classification performance are damaging [1].

2.2.4. Imbalance Types

An imbalanced dataset may suffer from one of the following problems, two or many of them. Imbalance could be either: **Binary** (Two-class) **Imbalance** (exist between the two classes) or **Multiclass Imbalance** (exist between more than two classes). It could also be considered **Between-Class Imbalance** (where the data sets revealing significant, and in some cases

extreme, imbalances of its distribution between the different classes) or **Within-Class Imbalance** (which concerns itself with the distribution of representative data for sub-concepts within a single class, i.e. in this case, a class is composed of many sub-clusters and some sub-clusters have much fewer examples than other sub-clusters. Although underrepresented sub-clusters can occur to both minority and majority classes, they are more likely to exist in the minority class, since it is often much easier to collect examples for the majority class). Imbalance may be **Intrinsic** (the imbalance is a direct outcome of the nature of the dataspace) or **Extrinsic** (the imbalance is not directly related to the nature of the dataspace. Variable factors such as time and storage causes the data imbalance [2]. For instance, suppose a data set is obtained from a continuous data stream of balanced data over an interval of time, and if during this interval, the transmission has irregular interruptions where data are not transmitted, then it is possible that the acquired data set can be imbalanced in which case the data set would be an extrinsic imbalanced data set attained from a balanced dataspace) [2]. Imbalance of data could be due to either **Relative Rarity** (Since class labels are essential to conclude the degree of class imbalance, class imbalance is typically assessed according to the training distribution. If the training distribution is representative of the underlying distribution, as it is often assumed, then there is no problem; but if this is not the case, then we cannot conclude that the underlying distribution is imbalanced, and we can say that its suffers from relative rarity) or **Absolute Rarity** (rare instances) (while class imbalance literally refers to the relative proportions of examples belonging to each class, the absolute number of examples available for learning is clearly very important. Thus, the class imbalance problem for a dataset with 10,000 positive examples and 1,000,000 negative examples is

clearly quite different from a dataset with 10 positive examples and 1000 negative examples, even though the class proportions are identical) [1].

2.2.5. Levels of Imbalance Solutions:

The solutions deal with the binary imbalanced learning problem can be divided into three major categories/levels: problem definition issues, data issues, and algorithm issues. **Problem definition issues** are very common. It takes place when having no enough information to accurately define the learning problem to solve it, or not owning the suitable metrics to evaluate the utility of the mined knowledge. The solution is easy but often not reachable: Redefine the problem in a simpler way for which more exact evaluation information is available and generate the suitable metrics to properly asses the mined knowledge after attaining the required knowledge or even suboptimal but good solutions. **Data issues** attend the actual data that is considerable for learning, and involves the problem of absolute rarity, where there are insufficient instances belong to one or more classes, to learn the decision boundaries associated with that class. The direct solution is to obtain additional training examples via Sampling techniques or Active Learning methods or other information acquisition strategies. Finally, **Algorithm issues** occur due to deploy an inadequate learning algorithm that performs poorly for imbalanced data such as applying an algorithm designed to optimize accuracy to an imbalanced learning problem, or the inability to discover indirect patterns – such as those belong to very rare classes- in data that may be hidden because of the data imbalance and the class imbalance (relative rarity). They also involve the incapability of handling data fragmentation issue. These issues could be handled by having an appropriate non-greedy search algorithm and not repeatedly partition the search space,

use a suitable evaluation metric to guide the heuristic search process as well as an appropriate inductive bias for imbalanced learning and deploy algorithms that explicitly or implicitly focus on the rare classes or rare cases or only learn the rare class [1].

2.2.6. Methods and strategies for dealing with imbalanced data

In the recent years, extensive efforts have been developed to handle the imbalanced data problem. They operate in the three levels (problem, data and algorithm levels) whether individually or cooperatively as hybrid approaches to better tackle the problem deploying diverse ways and strategies. This subsection review majority of rebalancing approaches and details their strategies [1]. Figure 2-1 summarizes them.

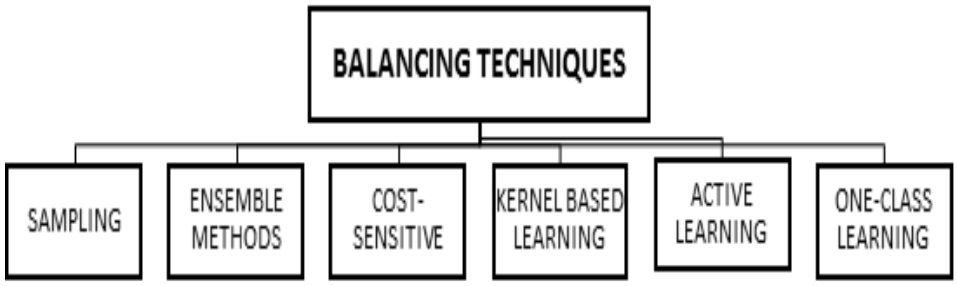


Fig. 2-1: Summary of Balancing Techniques

2.2.6.1. Sampling methods

Sampling methods employ many various approaches to modify the training set in such a way to generate more balanced amenable class distribution. One challenge with sampling strategies is deciding how much to sample, which is obviously conditioned on the sampling strategy that is deployed [1].The basic work in this area includes, Random Undersampling [3] where the majority class instances were discarded randomly until more balanced

distribution was reached. Here, vast quantities of majority data could be discarded making the decision boundary between minority and majority instances harder to learn, resulting in a loss in classification performance. So, in order to overcome these limitations, more sophisticated sampling techniques which were called Data Cleaning techniques had been developed to retain all useful information present in the majority class while removing redundant noisy, and/or borderline instances from the dataset such as Tomek Links that utilized by Kubat and Matwin et al. [4] and a modified version of the Condensed Nearest Neighbor (CNN) rule [5] to create a directed Undersampling Method. Another method for undersampling was proposed by Laurikkala et al. [6] which he called the Neighborhood Cleaning Rule (NCR). But there was a rudimentary old work was the Random Oversampling where minority class instances were copied and repeated in the dataset until a more balanced distribution was reached. Here, instances were repeated, (sometimes to very high degrees) in such a way that could cause drastic overfitting to occur in the classifier- cause the learned model might fit the training data too closely- resulting in declining the generalization ability of the classifier for the unseen data. In order to overcome this issue, Chawla et al. [7] developed a method of creating synthetic instances instead of just copying existing instances in the dataset. This technique was known as The Synthetic Minority Over-Sampling Technique (SMOTE). Basing on the feature space similarities between existing minority examples, it created artificial data altering the training set distribution by adding synthetically generated minority class instances, so the minority class became more balanced. Depending on its effectiveness, many extensions for it had been developed such as Adaptive Sampling methods like: Borderline-SMOTE, Adaptive Synthetic Sampling

(ADASYN) [8] algorithms, as well as Focused Resampling which was developed by Japkowicz et al. [9]. Some researchers performed a type of combination of the different previous methods. For instance, they included SMOTE+Tomek and SMOTE+ENN [10] where SMOTE was used to oversample the minority class, while Tomek and ENN, respectively, were used to under-sample the majority class.

2.2.6.2. **Ensemble learning**

It is an important paradigm in machine learning, it uses a set of classifiers to make predictions. The generalization ability of ensemble classifiers is generally much stronger than individual ensemble members. Basing on its simplicity, several ensemble methods had been integrated with sample methods in several ways that could be categorized into: **Bagging-style methods** such as UnderBagging [11] , OverBagging [11], SMOTEBagging [11], **Boosting-based methods** such as SMOTEBoost [12], RUSBoost [13], and DataBoost-IM [14], and **Hybrid ensemble methods** such as EasyEnsemble [15] and BalanceCascade [15]. It should be noted that many of the ensemble methods devoted for class imbalance learning were significantly better than standard ensemble methods and sampling-based class imbalance learning methods such as EasyEnsemble, and BalanceCascade [1].

2.2.6.3. **Cost-sensitive Learning Methods**

They use different cost matrices that describe the costs for misclassifying any specific data example instead of creating balanced data distributions as the sampling approaches strategy. Their methodology is grounded on the cost matrix concept which represents the numerical representation of the penalty of classifying examples from one class to another. They assign no

cost for correct classification of either the minority or the majority class considering the standard matrices. The cost of misclassifying minority examples should be higher than the contrary case. So, the objective of cost-sensitive learning is to generate a hypothesis that minimizes the overall cost on the training data set, which is usually the *Bayes conditional risk*. Generally, there are three categories of approaches to implement cost-sensitive learning for imbalanced data. The first class of techniques applies misclassification costs to the dataset as a form of dataspace weighting (translation theorem [16]; these techniques are basically cost-sensitive bootstrap sampling approaches where misclassification costs are used to select the best training distribution. The second class applies cost-minimizing techniques to the combination schemes of ensemble methods grounding on (Metacost framework [17]; this class consists of various meta techniques, where standard learning algorithms are integrated with ensemble methods to develop cost-sensitive classifiers - especially boosting-based methods- many of the existing research works in this area integrated the Metacost framework with dataspace weighting and adaptive boosting to get stronger classification results such as CBS1, CBS2 [18], and AsymBoost [19] which modified the weight-distribution-updating rule, so that the weights of expensive examples were higher. Some methods, such as linear asymmetric classifier (LAC) [20], changed the weights of the base learners when forming the ensemble. Some methods, such as AdaC1, AdaC2, AdaC3 [21], and AdaCost [22], not only changed the weight-updating rule, but also changed the weights of base learners when forming ensemble, by associating the cost with the weighted error rate of each class. Moreover, some methods directly minimized a cost-sensitive loss function, such as Asymmetric Boosting [23]. The third class of techniques incorporates cost sensitive

functions or features directly into classification paradigms to essentially "fit" the cost-sensitive framework into these classifiers such as the cost-sensitive decision trees [24], [25], cost-sensitive neural networks [26], [27], cost-sensitive Bayesian classifiers [28], and cost-sensitive support vector machines (SVMs) [29], [30], [31]. There is no unifying framework for this class of cost-sensitive learning because many of these techniques are specific to a particular paradigm, but in many cases, solutions that work for one paradigm can often be abstracted to work for others [1].

2.2.6.4. **Kernel-based Learning Methods**

The principles of kernel-based learning are based on the theories of statistical learning and Vapnik-hervonenkis (VC) dimensions [32]. The representative kernel-based learning paradigm, Support Vector Machines (SVMs), can provide relatively robust classification results when applied to imbalanced data sets [33]. SVMs enable learning by using specific examples near concept boundaries (support vectors) to maximize the separation margin (soft-margin maximization) between the support vectors and the hypothesized concept boundary (hyperplane), meanwhile minimizing the total classification error [32]. The effects of imbalanced data on SVMs exploit inadequacies of the soft-margin maximization paradigm [34], [35]. SVMs are inherently biased toward the majority concept, since they target minimizing the total error. There have been many studies that integrated kernel-based learning methods with general sampling, ensemble, cost-sensitive, active learning and standard classification techniques for imbalanced learning [2]. In addition, too many efforts tried to modify the SVMs kernel function in numerous ways as it will be detailed in section 2.5.

2.2.6.5. **Active Learning Methods**

Traditionally, Active Learning methods were used to solve problems related to unlabeled training data, then they had also been investigated in the community for imbalanced learning problems. Many studies integrated them with other approaches. They have been incorporated with sampling techniques by Zhu and Huang et al. [36] who analyzed the effect of undersampling and oversampling techniques with active learning for the word sense disambiguation (WSD) imbalanced learning problem. Also, Ertekin et al. [37], [38] suggested an efficient SVM-based active learning method that queries a small pool of data at each iterative step of active learning instead of querying the entire dataset [1].

2.2.6.6. The One-Class Learning Methods

Contrary to the standard classification methods that try to differentiate between instances of both positive and negative classes following discrimination-based inductive methodology, these methods aims at recognizing instances of a concept by using mainly, or only, a single class of examples (i.e., recognition-based methodology). Representative work in this area includes the one-class SVMs [39], [40] and [41].

2.3. Multiclass Classification

2.3.1. What Is Multiclass Classification?

It is a supervised multiclass classification algorithm where each training point belongs to one of ***N*** different classes, so the aim is at constructing a function that will correctly assign each new data point to the class it is belongs to. In other words, given a training data set of the form (x_i, y_i), where $x_i \in R_n$ is the i^{th} example and $y_i \in \{1. . . K\}$ is the i^{th} class label, Find a learning model **H** such that: $H(x_i) = y_i$ for new unseen examples [42].

Multiclass classification assumes that each sample is assigned to one and only one label: a fruit can be either an apple or a pear but not both at the same time.

2.3.2. Problems of Learning from Multi-Class Data

There are many problems appear when learning from this kind of data: In multiclass classification, several boundaries should be determined and constructed, this may lead to increase the probability of error. Moreover, Zhou and Liu [43] stated that most of the techniques developed for two-class problems become ineffective when dealing with multiclass learning problems and some methods are not applicable directly such as random oversampling and undersampling techniques. In addition, the performance evaluation metrics that dedicated for two class scenarios are not suitable for assessing the results of classification algorithms considering multiclass data accurately. These facts reveal the need for more sophisticated evaluation metrics.

2.3.3. Methods of Handling Multi-Class Data:

Initially, let us consider two traditional types of classification methods for Multiclass data: Flat Classification and Hierarchical one, The Flat Classification indicates to a single level of classes that examples should be assigned to one or more of them. On the other hand, The Hierarchical classification we intend refers to the existence of number of levels of classes where each example could be assigned to some at any level, meanwhile a hierarchical classification problem originally refers to a problem that involves a large number of classes, where some subsets of classes are more

closely related than others or where each node is the sub-class of its parent's node. Therefore, these methods – Hierarchical methods for classification- are subdivided into Hierarchical Classifiers and hierarchical Decomposition methods regarding the relations between classes [44] as it will be detailed latter in this section.

Hierarchical models for classification suffer from the difficulty of making many decisions prior to obtain the final classification result. This intermediate decision making leads to the error propagation phenomenon causing a decrease in accuracy. On the other hand, although flat classifiers are based on a single decision including all the final classification results, it is difficult to make a single decision as it involves many results, which is probably unbalanced. [45]

There are three methods of modeling a multi-class pattern classification problem that could be kinds of flat classification- noticing that they could be utilized also in a level or more of the levels of the hierarchical models-: Extensible algorithms, Class Decomposition methods and Error-Correcting Output-Coding (ECOC), meanwhile, there are three types of hierarchical structures that belong to the hierarchical models for classification of Multiclass data, as clarified in Figure 2-2.

In this section, we will review these techniques and solutions introduced in this area because they form main parts of the solutions of tackling the multiclass imbalanced learning problem.

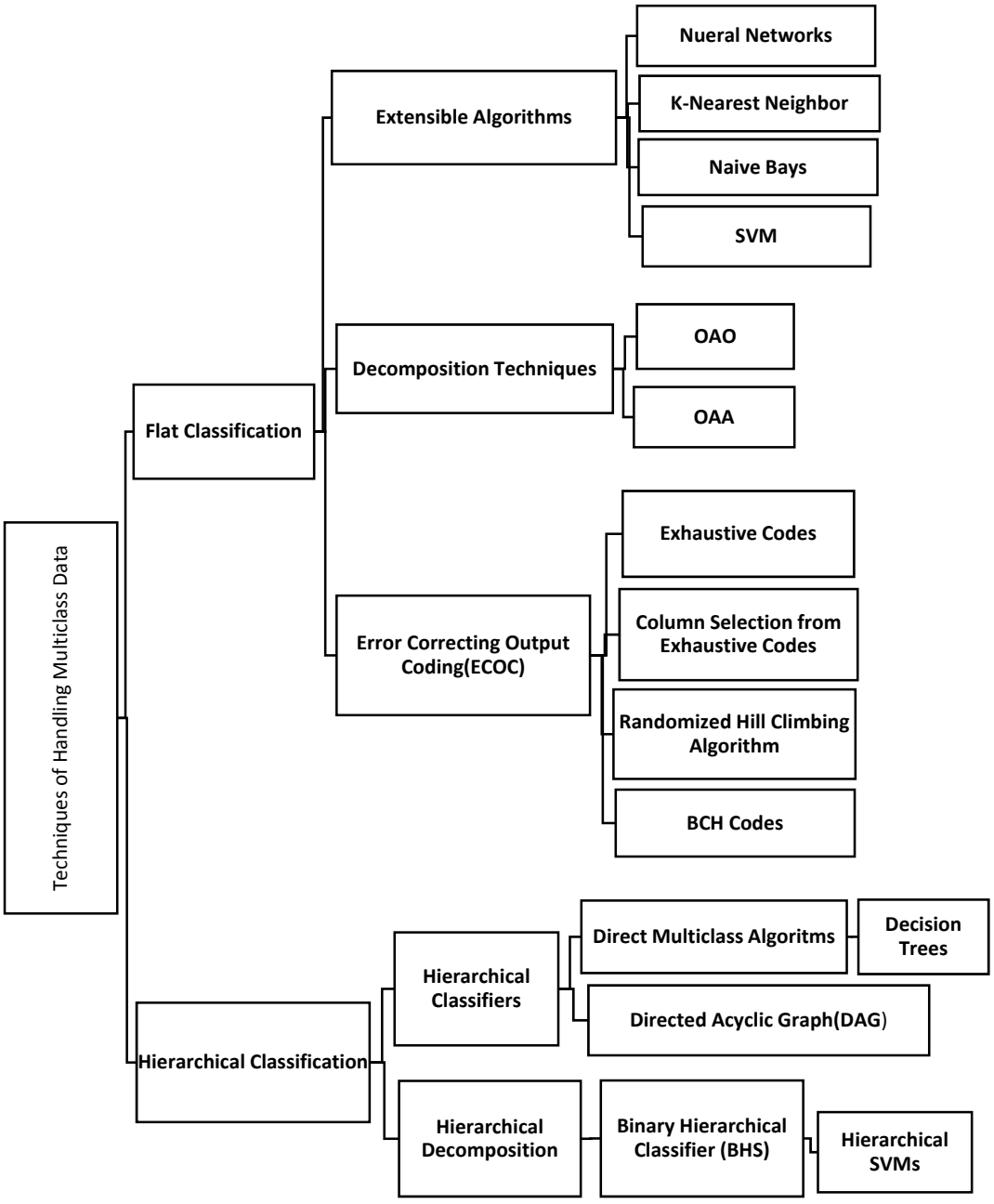


Fig. 2-2:Summary of Multiclass techniques

2.3.4. **Extensible algorithms**

They are problem adaption techniques, rely on extending binary classification problems to manipulate multiclass data directly by adapting

some specific algorithm without any class decomposition. They include Decision Trees, Neural Networks, k-Nearest Neighbor, Naive Bayes classifiers, and Support Vector Machines [46] which will be detailed in this chapter later.

2.3.5. Class Decomposition Approaches:

They are called (Binarization techniques) as well [47].They are problem transformation techniques that solve a problem by breaking it up into smaller ones and solving each of the smaller subproblems separately utilizing an independent binary classifier for each sub problem [48]. Then, the results of the binary classifiers are combined to get the classification result [49] . However, this is not necessarily the best approach for certain application problems. Decomposing a big problem has several advantages: firstly, individual classifiers are likely to be simpler than a classifier learns from the whole data set and exchanging one of them will not conflict the others. Secondly, they can be trained independently to allow various feature spaces, feature dimensions and architectures instantaneously for less modeling time. A possible drawback could be existed when each individual classifier is trained without full data knowledge, causing classification ambiguity or uncovered data regions with respect to each type of decomposition [50]. Here are the main schemes that majority of the state-of- art solutions are based on:

2.3.5.1. One-Against-All (OAA) scheme:

It is also called One-vs-All (OVA) decomposition method. Given a c-class decomposition task ($c > 2$): OAA constructs c binary classifiers, a classifier is constructed for each class. So, each problem is faced up by a binary classifier which is responsible of distinguishing one of the classes from all

other classes. To train the classifiers the whole training data is used, considering the patterns from the single class as positives and all other examples as negative (this can cause imbalanced training data, and if the original data is imbalanced already or includes a large number of classes, so the problem will be worse). In other words, a classifier F_i is trained using the samples of class C_i against all the samples of the other classes. In testing phase, a pattern is presented to each one of the binary classifiers, the classifier which gives a positive output indicates the output class. In many cases, the positive output is not unique so, some tiebreaking technique must be applied, a decision function that assigns the test sample to the class with the highest output value among all [51] can be utilized.

In addition to its computational efficiency (only N classifiers are needed), another advantage of this approach is its interpretability. Since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier. This is the most commonly used strategy and is a fair default choice. But uncovered and overlapped regions in the data space could be found because the classification boundary produced by each individual classifier is independent from those of other classifiers [50].

2.3.5.2. One-Against-One (OAO) scheme:

This approach constructs one classifier per pair of classes, each of the c classes is trained against every one of the other classes. It results in $\boldsymbol{c}\,(c-1)/2$ binary classifiers. The training of the classifiers is done using as training data only the instances from the original dataset which output class is one of both classes, instances with different output classes are ignored. In validation phase, a pattern is presented to each one of the binary classifiers.

A combining strategy of their outputs is necessary for a final decision. The simplest way is a majority vote, so the test sample is assigned to the class with the highest number of votes [52]. Also, Classification by Pairwise Coupling (PC) is utilized as combination strategy and it enhance some shortcoming of the former strategy [53]. Major advantages of OAO is that it consolidates the prediction of each class, as well as generalization performance. If one classifier makes a classification mistake, others still have chance to make it up. Also, it has not to produce imbalanced training data while owning the ability of incremental learning. When a new class joins to the current data, we just need to build another c new classifiers without affecting the existing ones. However, this approach suffers from some disadvantages [54]: The number of individual classifiers grows fast in a quadratic rate of c. When c is large, the training time can be very long. In addition, this method is usually slower than one-vs-the-rest due to its complexity. In terms of the imbalanced learning, the performance of OAA and OAO s are highly hindered by the imbalance presence despite the popularity and the successful utilization in different domains of them [50].

2.3.6. Error-Correcting Output-Coding (ECOC):

This approach incorporates the idea of error-correcting codes [55], which was designed originally to correct errors during data transmission for communication tasks by exploring data redundancy. This approach represents the machine learning task as a kind of communications problem in which the identity of the correct output class for a new example is being transmitted over a channel. The channel consists of the input features, the training examples, and the learning algorithm. The class information is corrupted due to errors caused by the poor choice of input features, the finite

training sample, and flaws in the learning process. The system may mend from these errors by encoding the class in an error-correcting code and transmitting each bit separately (i.e., via a separate run of the learning algorithm). So, ***N*** binary classifiers are trained to distinguish between the ***K*** different classes where each class is given a codeword of length ***N*** according to a binary matrix ***M***. Each row of ***M*** corresponds to a certain class. Each class is given a row of the matrix. Each column is used to train a distinct binary classifier. When testing an unseen example, the output codeword from the ***N*** classifiers is compared to the given ***K*** codewords, and the one with the minimum hamming distance is considered the class label for that example. Diettrich and Bakiri [55] reported improved generalization ability of this method over the above two techniques.

A measure of the quality of an error-correcting code is the minimum Hamming distance between any pair of code words which is the different bits between two codewords. For instance; if a 7-bit codewords are associated with classes C1…C4. Given unknown tuple X, supposing the 7-trained classifiers output is 0001010. This will mean that:

H (X, C1) = 5, by checking # of bits between [1111111] & [0001010]

H (X, C2) = 3, H (X, C3) = 3, H (X, C4) = 1, thus C4 is as the label for X

Error-correcting codes can correct up to ((h-1)/h) 1-bit error, where h is the minimum Hamming distance between any two codewords and If we use 1-bit per class, this indicates to its equivalence to one - vs.-all approach which leads to the fact that the codes are insufficient to self-correct, so when selecting error-correcting codes, there should be good row-wise and column

wise separation between the codewords, or another method to construct error –correcting output could to be utilized.

There are four methods for constructing good error-correcting output codes were deployed in the state of art solutions for Multiclass learning considering the ECOC method: (a) an exhaustive technique, (b) a method that selects columns from an exhaustive code, (c) a method based on a randomized hill-climbing algorithm, and (d) BCH codes. The choice of which method to use is based on the number of classes k. Finding a single method suitable for all values of k is an open research problem [55].

2.3.7. Hierarchical Classification

Regarding the previously mentioned study of Beyan & Fisher et al. [44], the Hierarchical methods for classification could be one of two categories:

- **Hierarchical Classifiers** where the classes were organized in a pre-defined hierarchy like a tree. The tree is created such that the classes at each parent node are divided into many clusters, one for each child node. The process continues until the leaf nodes contain only a single class. At each node of the tree, a simple classifier, usually a binary classifier, makes the discrimination between the different child class clusters. Following a path from the root node to a leaf node leads to a classification of a new pattern.
- **Hierarchical Decomposition** where the factors like similarity of data form the class hierarchy [56]. Here, there is no pre-defined class hierarchy. Instead, this approach is based on placing the classes in a tree, usually a binary tree, utilizes a hierarchical division of the output space [46]. The most generic form of hierarchical

decomposition is dividing a multiclass problem in a hierarchical way to obtain binary hierarchical classifier.

Considering the first type, two methods follow its principle:

2.3.7.1. **Direct Multiclass:**

It's the Decision-tree algorithms that can be easily generalized to treat these multiclass learning tasks. Labeling each of the decision tree leaf can be done using one of the **k** classes, and the classification process of these classes can be carried out by selecting the internal nodes [55].

2.3.7.2. **The Decision Directed Acyclic Graph (DDAG)**

The Decision Directed Acyclic Graph constructs a rooted binary acyclic graph where each node is associated to a list of classes and a binary classifier. The root node considers all classes in the list and one classifier distinguishing between two of the classes (generally, the first and the last). According to the prediction of the classifier, the class which has not been predicted by the classifier is removed from the list and a new node is reached (the node associated to the new list, which also has another binary classifier discriminating between the first and the last classes from the new list). The last class remaining on the list is the final output class [57]

Regarding the second type of the hierarchical methods which is Hierarchical Decomposition, the following trees is an interesting example for approaches that follow its principle:

Binary Tree of Classifiers (BTC) or Binary Tree of SVM (BTS), easily can be extended to any type of binary classifiers that decreases the number of

classifiers and increases the global accuracy using some of the binary classifiers which distinguish between two classes to simultaneously discriminate other classes. The tree is constructed recursively and in equivalent way to DDAG approach, each node has associated with a binary classifier and a list of classes, but in this case, the decision of the classifier can distinguish other classes as well as the pair of classes used for training. So, in each node, when the decision is done, more than one class can be removed from the list. In order to avoid false assumptions, a probability is used when the examples from a class are near the boundary so the class cannot be removed from the lists in the following level.

In this technique, a hierarchy can be created using the similarity of classes**,** for instance, Kumar et al. [58] organized classes in a hierarchy collecting similar classes together to transform the multi-class classification problem into a binary classification problem. For text mining, SVM based hierarchical clustering was used utilizing the similarities between features [59].

Chen et al. [60] use a similar approach of clustering the classes into a binary tree called Hierarchical SVM (HSVM). However, the clustering is performed via arranging classes into an undirected graph, with edge weights representing the Kullback-Leibler distances between the classes, and split the classes into two sub-clusters that are most distant from each other. SVMs are used as the binary classifier at each node of the tree. They reported improved performance versus bagged classifiers using remote sensing data.

2.4. Imbalance Multi-Class

2.4.1. What is Multi-Class Imbalance Problem?

The imbalanced multiclass problem belongs to supervised machine learning tasks where each instance should be assigned to one of ***N*** different classes with unequal sample sizes.

It's obvious that the effect of the presence of both problems in the data – being imbalance and multiclass - is more severe and rises the need for more analysis and investigation, since some techniques that are applicable for balanced multiclass data are not if they applied over imbalanced multiclass data.

2.4.2. Methods of Handling Multiclass Imbalanced Data:

These solutions naturally were emanated from those which dedicated for treating the binary imbalanced data and those for the multiclass ones. So, they also could be subjoined to the traditional types of classification methods for Multiclass data: Flat and Hierarchical Classification methods.

Regarding Flat classification, there are two main methods – Figure 2-3 - followed in the research community to deal with such data:

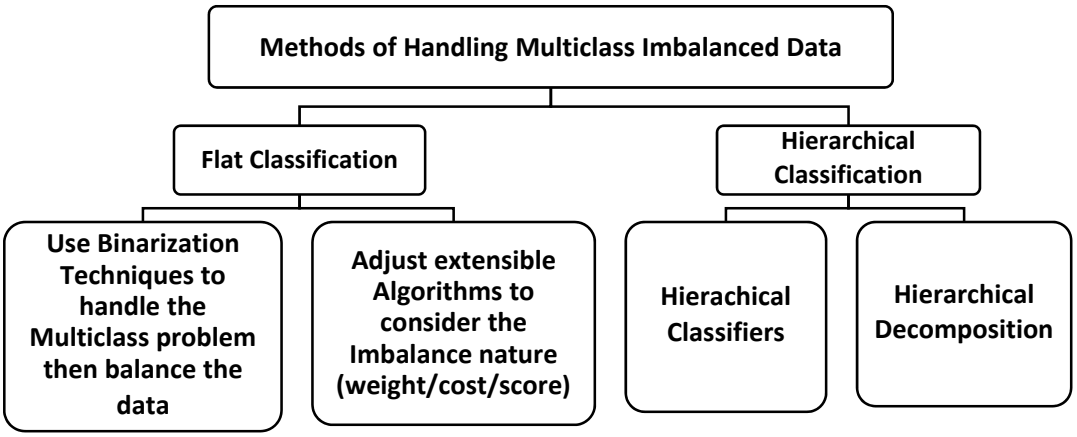


Fig. 2-3Methods of Handling Multiclass Imbalanced Data

2.4.2.1. **Using Binarization Techniques**

The first and the oldest approach developed to treat such data, which is using Binarization techniques to transform the multiclass nature of the data into binary imbalanced sub datasets, then in subsequent step, rebalance the binary sub datasets using one of balancing techniques which were indicated in Figure 2-3 to be able to start classification process.

Fernandez et al. [61] integrated OAO with SMOTE in their algorithm. Instead of using data-level methods in two steps: firstly, they deployed OAO. Then, whenever each one of these binary subproblems is imbalanced, an oversampling using the SMOTE algorithm was exploited before the pairwise learning process. Wang and Yao et al. [62] studied the effect of two kinds of multi-class imbalance problems; multi-minority and multi-majority on the performance of two basic resampling techniques. They both showed strong negative effects. Then they applied AdaBoost.NC to several real-world multi-class imbalance datasets and compared it to other three popular

ensemble methods based on the correlation analysis and performance pattern analysis ensemble methods. AdaBoost.NC was better at recognizing minority class examples and balancing the performance among classes in terms of G-mean without using any class decomposition, meanwhile, using class decomposition (the one-against-all scheme in their experiments – OAA) did not provide any advantages in multi-class imbalance learning in their experiments. On other hand, Chen & Lu et al. [60] proposed an algorithm that used OAA, then they depended on sampling methods to further decompose each binary problem and rebalance the training set. Zhao & Li et al [63] used OAA in addition to undersampling and SMOTE techniques to remedy the imbalanced distribution in their protein data. Similarly, to Wang study, Choon & Gilbert et al. [64] proposed utilizing ensemble methods for classification as well. They combined the eKISS Method rules of base classifiers to generate new classifiers. They had applied the PART rule-based machine learning technique to generate the base classifiers for their ensemble learning system to improve the coverage of examples from small protein classes. Then they deployed both OAA and OAO schemes to generate one new classifier per class, called the ensemble classifiers. Ghanem & Venkatesh et al. [65] suggested a method called Multi-IM which derived its fundamentals from the probabilistic relational technique (PRMs- IM) that was designed for learning from imbalanced relational data for the two-class problem, in addition to All-and-One (A&O) approach to treat the imbalanced problem. Then an independent classifier was trained on each balanced subset. They used the weighted voting strategy as applied in PRMs-IM to combine classifiers to get the result for the parent classifier. Liao et al. [66] investigated variety of oversampling and undersampling techniques used with OAA for a weld flaw classification

problem in addition to three algorithms including minimum distance, nearest neighbors, and fuzzy nearest neighbors that were utilized as the classifiers. Abdi & Hashemi et al. [67] combined over-sampling (Mahalanobis distance-based over-sampling technique -MDO in short-) into boosting algorithm and called it MDOBoost. They over-sampled the minority classes via MDO considering the original minority class characteristics. MDO generated more similar minority class examples to original class samples more than SMOTE. The study of Platt & Cristianini et al. (57) didn't consider the Binarization technique for handling the Multiclass situation, instead, they deployed a balancing technique (Dynamic sampling method (DyS)) for multilayer perceptrons (MLP) to deal with the multiclass nature of the data, then combined the outputs of the ensemble as multi-class classifier. This study utilized the idea of using Codewords beside OAA. Jeatrakul et al. [68] suggested the One-Against-All technique with Data Balancing (OAA-DB) algorithm which was an extension of OAA and aimed at improving the weakness of OAA. It balanced the data utilizing combination of SMOTE and CMTNN and combined it with OAA. CMTNN worked as an under-sampling technique while SMOTE was applied as an over-sampling technique. The multi-binary classifier generated $\boldsymbol{K}$ outputs of $\boldsymbol{K}$ classes, each $\boldsymbol{K}$ output was converted to a binary bit to produce binary codewords of each testing example. A binary codeword was represented by the $\boldsymbol{K}$ bits class output of each testing instance to utilize it in the classification process. Alejo et al. [69] algorithm made the error function of neural networks cost-sensitive by incorporating the proportion of classes within the data set to confirm minority classes, after OAA was applied.

2.4.2.2. **Adjust the Extensible Algorithms:**

The second approach followed in the research community to handle imbalanced multiclass data is adjusting the Extensible Algorithms [46] to consider both imbalance and multiclass problems. Here, the modification introduces costs into classification process or moving decision threshold. This can be applied by utilizing cost sensitive methods to find an appropriate cost matrix with multiple classes and suit its imbalance nature such as these following studies:

Langford et al [70] combined two ideas; firstly, to enhance the performance of neural network on multiclass imbalanced data, he deployed diverse random subspace ensemble learning with evolutionary search. In order to increase the performance of the learning and optimization of neural network, he exploited the minimum overlapping mechanism to provide diversity. Secondly, to optimize the misclassification, an evolutionary search technique was utilized cost under the guidance of imbalanced data measures. Some studies assigned different misclassification costs through using SVMs classifier. The misclassification cost of the minority classes must be higher than the majority class's. So, SVMs could handle all imbalanced multiclass data in one optimization formulation. For instance, the study of Landgrebe and Duin et al [71] who proposed a multi-class Weighted Support Vector Machines (WSVM) method to perform automatic recognition of activities in a smart home environment. This method supported analytic parameter selection of the + C and − C regularization parameters with a new criterion from the training data directly, based on the proportion of class data. In empirical study Wei & Lin [72] compared the performance of MultiSVM that considered all classes at once with three methods based on binary

classifications: “one-against-all,” “one-against-one,” and directed acyclic graph SVM (DAGSVM). They concluded that the “one-against-one” and DAG methods are more suitable for practical use.

Ensemble algorithms and Boosting techniques that modify the weight updating rule and/or loss function such that the minority examples were emphasized with higher weights, or high scores for most interested and confident instances were deployed as well.

There were other studies that utilized the cost sensitive method to rebalance the multiclass data. They generally categorized the costs into two types: **Example-dependent cost** which assumed that each example had its own misclassification cost, and **Class-dependent cost** which assumed that each class had its own misclassification cost [27]. According to Zhou [27] , he recommended investigating the consistency of the costs to utilize the rescaling approach firstly. He suggested that applying rescaling after decomposing the multi-class problem into a series of two-class problems is better if the cost is not consistent. Wang et al. [50] introduced a typical study to utilize the misclassification cost with ensemble classifiers as well.

2.4.2.3. **Hierarchical Classification:**

Generally, the hierarchical classification techniques which are dedicated for treating data that suffers from both problems- Imbalance and Multiclass- handle the imbalance nature initially, then lever the multiclass situation by turning the classification process into stages of levels.

Regarding **Hierarchical Classifiers,** One-Against-Higher-Order (OAHO) method [73] stood on a hierarchy of classifiers based on the data distribution. OAHO constructed ***K-1*** classifiers for ***K*** classes in a list of {C_1,

$C_2 \ldots C_K\}$. The first classifier was trained using the samples of the first class C_1 against all the samples of all the other classes. Then, the second classifier was trained using the samples of the second class in the list C_2 against the samples of the higher ordered classes $\{C_3 \ldots C_K\}$ and so on until the last classifier was trained for C_{K-1} against C_K. To diminish the imbalanced situation, the classes were organized descendly according to the number of the samples in each class, in which the small classes were grouped together against the majority class. The problems were that misclassifications made by the top classifiers couldn't be improved by the lower classifiers and OAHO performance was sensitive to the classifier order. Li et al. [74] suggested automatic music genre classification approach where the taxonomy gave the relationship between the genres and the similarity matrix from linear discrimination was utilized to construct automatic taxonomies. Wu et al. [75] constructed a tree for handling the multi class nature of the data and a multiclass classifier at each parent node.

Considering **Hierarchical Decomposition**, it splits the multiclass problem in a hierarchical way such that binary hierarchical classifiers could be used. For instance, Cesa-Bianchi et al [56] utilized the similarity of classes to construct a hierarchy. Also, considering the study of Ramanan et al. [76] in which they proposed the learning architecture (Unbalanced Decision Tree (UDT)) standing on Directed Acyclic Graph (DAG) and One-versus-All (OVA) approaches. At each decision node, The OVA based concept was implemented. Each decision node of UDT was considered an optimal classification model. The based classifier of the OVA which resulted the maximum performance measure was considered the optimal model for each decision node. Beginning with the root node, the optimal model evaluated

one selected class against the rest. Then, from the level of the decision tree, the UDT removed the selected class moving to the next level. When the algorithm yields an output pattern it terminated at a level of the decision node. Also, hierarchical SVM was proposed by Chen, Crawford and Ghosh et al. [77]. Basing on class similarities the classes were partitioned into two subsets until one class label was found at a leaf node. Moreover, in the previously mentioned study of Beyan & Fisher et al. [44], they presented a hierarchical decomposition method which based on clustering and they deployed outlier detection for classification. The hierarchy grounded on the similarities of data (i.e. clusters). Different data and feature subsets where employed to construct the hierarchy levels. Supposing that the minority class samples in each class were outliers by cardinality, or by their distance to class, Classification of minority class samples was done via Outlier detection center. Hoens, Chawla and Zhou et al. [78] suggested using Hellinger distance decision trees (HDDTs) to solve the class imbalance problem for decision trees without sampling. They compared different methods of building C4.4 and Hellinger distance decision trees for multi-class imbalanced datasets. Luo et al. [79] proposed a hierarchical classification method which was a simple bi-classifier with less features input made out most normal samples with an allowable low error rate for minority samples, then a complicated multi-classifier with more features input was constructed by learning the rest less imbalanced samples. To get accurate output for every class, they deployed complicated classifier of ANN ensembles. For classification process, two classifiers operated in parallel. When normal-class result had been acquired the simple classifier of the first layer was able to end the second one.

2.4.3. An Abstract comparison between Multiclass Imbalanced Solutions

- **ADVANTAGES:**

Naturally, the Pros and cons of each method are originated from the characteristics of each techniques that forms a part of the whole strategy of treating the Multiclass imbalanced problem. For instance, SVMs is a very strong algorithm that has big generalization capability and strong mathematical background, so it works very well, even with very small training sample sizes comparing with Binarization techniques, but according to Wei & Lin [72] the later techniques are more suitable for practical use, specifically when dealing with large scale problems and they are more accurate for rule learning algorithms.

Considering the hierarchical decomposition, dividing the problem into smaller problems by the hierarchy results in selecting a smaller set of features (a more specific domain term features) to a sub-problem which increases the accuracy and efficiency.

Many Studies such as [59], [80], [81], [82] agreed that comparing hierarchical methods to Flat classification techniques, the former can have better classification results.

- **DISADVANTAGES:**

The Binarization approach suffers from excessive testing time because of the need of combing the results of $\boldsymbol{k}$ ($\boldsymbol{k}$-1)/2 binary classifiers.

Adding weights or scores modifying the kernel functions of the extensible algorithms faces the difficulty of constructing direct connections between the parameters. Moreover, during training time, a matrix of kernel values for

every pair of examples must be computed noticing that SVM is slow and owns computational complexity in training according to its nature - the hyperplane it deals with and its kernel function -, so regarding large-scale problems, learning can take a very long time when dealing with MultiSVM with scores.

Using the hierarchical approaches rises the need to proceed until a leaf node is reached to decide on any input pattern, so it also consumes time depending on the path.

Digging deeper, the characteristics of the dataset affects directly on how to decide the most suitable solution to handle each part of the problem of the data nature - Multiclass or imbalance- for any considering learning problem: The number of instance whether its large scale or small one, the number of its classes and number of attributes, the degree of the imbalance in instances distribution and other data complexity if exists.

2.5. Kernel-Based Learning Methods (Support Vector Machines) for Class Imbalance Learning

Since the proposed hierarchical model will be build utilizing the Support Vector Machine (SVM) and Multi SVMs, it's important to stand on their structure and details.

2.5.1. Background

Support Vector Machine is a tool for machine learning to solve the problem reorganization problem. It also has been effectively applied to many real-world classification problems from various domains due to its theoretical and practical properties, such as solid mathematical background, high generalization capability, the ability to obtain global and nonlinear

classification solutions using different kernel functions and good performance comparing to other classifiers. SVMs can manage many dimensions as well as providing good classification results in case of small sizes of training samples. Moreover, it has ability to process many thousand different inputs which is very critical aspect in some real-life application such as text classification, because it opens the opportunity to use all words in a text directly as features regardless to how long is it [83].

2.5.2. Introduction to the basic Support Vector Machines

We briefly review the learning algorithm of SVMs, which has been initially proposed in [32], [84] and [85]. In a binary classification problem represented by a dataset $\{(x_1, y_1), (x_2, y_2) \ldots (x_l, y_l)\}$, where $x_i \in R^n$ represents an n-dimensional data point, and y_i $\{-1, 1\}$ represents the label of the class of that data point, for i = 1. . . l. The goal of the SVM learning algorithm is to find the optimal separating hyperplane that effectively separates these data points into two classes. In order to find a better separation of the classes, the data points are first considered to be transformed into a higher dimensional feature space by a nonlinear mapping function ƒ. A possible separating hyperplane residing in this transformed higher dimensional feature space can be represented by:

$$\boldsymbol{w} \cdot f(\boldsymbol{x}) + \boldsymbol{b} = \mathbf{0} \qquad (\mathbf{2.1})$$

where **w** is the weight vector normal to the hyperplane. If the dataset is completely linearly separable, the separating hyperplane with the maximum margin (for a higher generalization capability) can be found by solving the following maximal margin optimization problem:

$$\boldsymbol{min\ (\tfrac{1}{2}\, w \cdot w)}$$

S.t. $yi\,(w \cdot f(x) + b) \geq 1$ (2.2)

$i = 1, \ldots, l$

However, in most real-world problems, the datasets are not completely linearly separable even though they are mapped into a higher dimensional feature space. Therefore, constrains in the optimization problem mentioned in Equation (2.2) are relaxed by introducing a set of slack variables, $\xi i \geq 0$. Then, the soft margin optimization problem can be reformulated as follows:

$$\boldsymbol{min}\ (\frac{1}{2} w.w + C \sum_{i=1}^{l} \xi i\ )$$

S.t. $yi\,(w \cdot f\,(xi) + b) \geq 1 - \xi i$ (2.3)

$\xi i \geq 0,$ $i = 1, \ldots, l$

The slack variables $\xi i > 0$ hold for misclassified examples, and therefore the penalty term $\sum_{i=1}^{l} \xi i$ can be considered as a measure of the number of total misclassifications (training errors) of the model. This new objective function given in Equation (2.3) has two goals. One is to maximize the margin and the other one is to minimize the number of misclassifications (the penalty term). The parameter $\boldsymbol{C}$ controls the trade-off between these two goals. This quadratic optimization problem can be easily solved by representing it as a Lagrangian optimization problem, which has the following dual form:

$$\textbf{Max}_{\alpha i} \left\{ \sum_{i=1}^{l} \alpha i - \frac{1}{2} \sum_{i=1}^{l} . \sum_{j=1}^{l} \alpha i\ .\ \alpha j\ .\ yi\ .\ yj . f\,(xi) . f\,(xj) \right\} \quad (2.4)$$

S.t. $\sum_{i=1}^{l} yi\ \alpha i = 0$. $0 \leq \alpha i \leq C.$ $i = 1. \ldots l$

where αi are Lagrange multipliers, which should satisfy the following KarushKuhn–Tucker (KKT) conditions:

$$\alpha i(yi(w.\int(xi)+b)-1+\xi i)=0. \quad i=1.\ldots l \qquad (2.5)$$

$$(\text{C-}\alpha i)\,\xi i=0. \quad i=1.\ldots l \qquad (2.6)$$

An important property of SVMs is that it is not necessary to know the mapping function ∫ ***(x)*** explicitly. By applying a kernel function, such that $K(x_i , x_j) = \int(x_i) , \int(x_j)$, we would be able to transform the dual optimization problem given in Equation (2.1) into Equation (2.4)

$$\textbf{Max}_{\alpha i}\left\{\sum_{i=1}^{l}\alpha i-\frac{1}{2}\sum_{i=1}^{l}.\sum_{j=1}^{l}\alpha i\,.\alpha j\,.yi\,.yj.\mathrm{K(xi\,.xj\,))}\right\} \qquad (2.7)$$

$$\text{S.t.}\quad \sum_{i=1}^{l}yi\,\alpha i=0\ . \qquad 0\le\alpha i\le C. \qquad i=1.\ldots l$$

By solving Equation (2.7) and finding the optimal values for αi, w can be recovered as in the following equation:

$$w=\sum_{i=1}^{l}\alpha i.\,wi.\,f(xi) \qquad (2.8)$$

b can be determined from the KKT conditions given in Equation (2.5). The data points having nonzero αi values are called Support Vectors. Finally, the SVM decision function can be given by:

$$f(x)=\sin(w.\,f(x)+b)=\sin(\sum_{i=1}^{l}\alpha i.\,yi.\,K(xi.\,x)+b) \qquad (2.9)$$

2.5.3. Multiclass Support Vector Machine:

Support Vector Machines (SVMs) were originally designed for binary classification. But currently, they are extended to deal with multiclass

problems and known as Multiclass Support Vector Machine (Multiclass SVM). Multi-SVMs are still an ongoing research issue.

Many strategies were introduced to utilize Multiclass SVMs, the most known are the following: The **first** traditional method utilized the concept of solving several binary classifications problems via one of three Decomposition techniques. It Started by constructing several OAA Support Vector Machine classifiers, then picking the class which classifies the test datum with greatest margin, or by building a set of OAO classifiers, then choose the class that is selected by the most classifiers, or deploying the Directed Acyclic Graph Support Vector Machines (DAGSVM) which proposed in [86], where the training phase proceeds in the same way as the one against-one method by solving k (k − 1)/2 binary SVMs. However, in the testing phase, it uses a rooted binary directed acyclic graph which has k ($\boldsymbol{k}$ − 1)/2 internal nodes and k leaves. Each node is a binary SVM of i^{th} and j^{th} classes. Given a test instance $\boldsymbol{x}$, beginning from the root node, an evaluation for the binary decision function is $\boldsymbol{c}$. so, depending on the output value depending on the output value it moves to either left or right. Therefore, to obtain the predicted class we go through a path before reaching a leaf node which indicates it. An advantage of using a DAG is that [86] some analysis of generalization can be established [72].

The second strategy was introduced by Kindermann et al. [83] . He investigated the influences of multiclass error correcting codes on the performance of the SVM over a wide variety of frequency code such as relative frequencies and logarithmic frequencies deploying many SVM kernel combinations.

The third strategy works more directly than the former. To be constructed, a larger optimization problem is needed, because it considers all data in one optimization formulation which make it computationally more expensive for multiclass problem to be solved than a binary problem due to its complexity for practical implementations Moreover, up to now experiments are limited to small data sets. Here, the SVM must own a different algorithm from the original SVM. So, many studies proposed different families of All-together Multiclass SVM, that vary in their kernel functions or its different parameters [72] such as [87], [88], [89] and [90]

Chih-Chung Chang and Chih-Jen Lin et al [91] introduced a library for SVM which they called LIBSVM. It is utilized by Rapidminer software that we depended on to carry out our experiments in this study. It does not support one-versus-one multi-classification, instead it deploys one-versus-all method. So, it if k is the number of classes, we generate $\boldsymbol{k}(\boldsymbol{k}\text{-}1)/2$ models, each of which involves only two classes of training data. According too them, inspite of the huge space needed to store $\boldsymbol{k}(\boldsymbol{k}\text{-}1)/2$ models thier implementation stores models in a sparse form and can effectively handle some large-scale data.

2.6. Summary:

This chapter analyzed the considered problem notion which is originated from other two problematic notions. It detailed each and reviewed the related literature, regarding their effects, challenges, the state-of art solutions – to our knowledge – and possible opportunities to stand on a good base to evaluate them and find the gaps, then conduct our own solution. In general, we had noticed that the introduced techniques suffer from being complex, from the perspective of consuming time in training or testing or even both.

The implementation of each solution is subject to many factors such as the utilized algorithm for classification, the data characteristics. Some solutions were successful when deployed in certain circumstances, but they failed in others. So, there is no one ideal algorithm for all cases.

3. CHAPTER THREE: Research Methodology

3.1. Introduction

This chapter discusses the research methodology and activities carried out to accomplish the research objectives. It details the steps followed to develop the hierarchical proposed SVM model of classification of our considering data as well.

3.2. The Research Phases:

Figure 3-1 explains the proposed phases towards achieving the research aim and objectives.

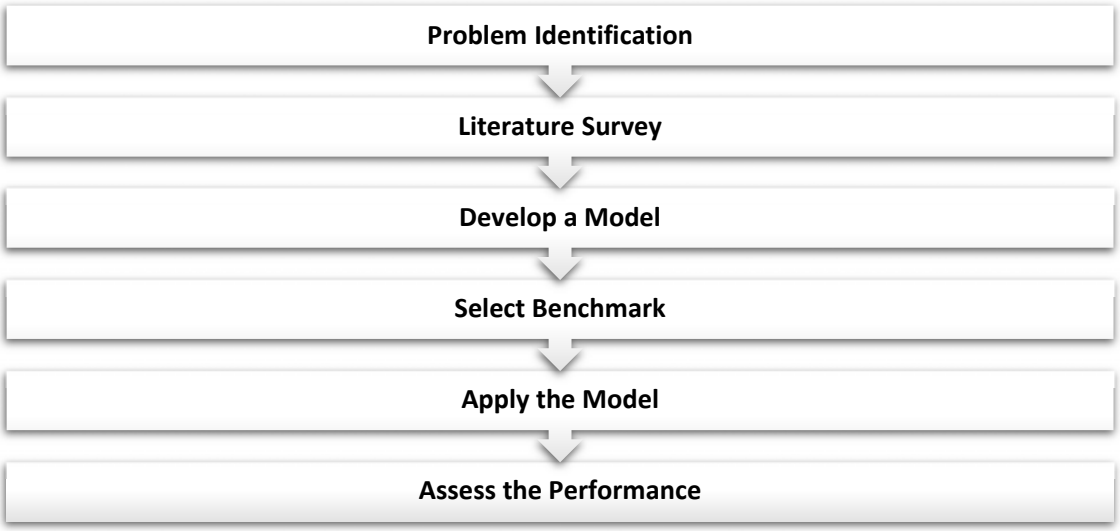


Fig. 3-1: Research Phases

3.2.1. Phase 1: Problem Identification

It was mentioned previously, that this research is concerning in building an optimal classification model for imbalanced multiclass data that generates precise classification results for the minority instances.

3.2.2. Phase Two: Literature Survey

In this phase, we read and analyzed more than hundred scientific papers and good related references to develop a solid background of classification of Imbalanced Multiclass task. The solutions introduced to tackle this problem were integrated ones. They are combination techniques based on the methods that were dedicated to handle the binary imbalanced data and those used to treat the multiclass cases. In other words, no direct dedicated solutions for treating our considering data. So, the second phase passed through three stages: the first one was reviewing the binary imbalance classification state- of – art solutions. The second one was exploring the multiclass classification ones and the third stage was investigating the methods that treat both situations. Finally, the SVM and MultiSVM machine structures were detailed. As a result of the literature survey, a summary of different solutions was conducted and compared to find a gap they didn't fill and better build the classification model.

Therefore, the first two objectives were accomplished through the previous two chapters.

3.2.3. Phase three: Develop the Model

Regarding the third objective, the following sub-section illustrates the structure of the proposed hierarchical model. Figure 3-2 shows an example for a dataset that is consists of six imbalanced classes and clarifies the steps that it passes through when applying the model over it.

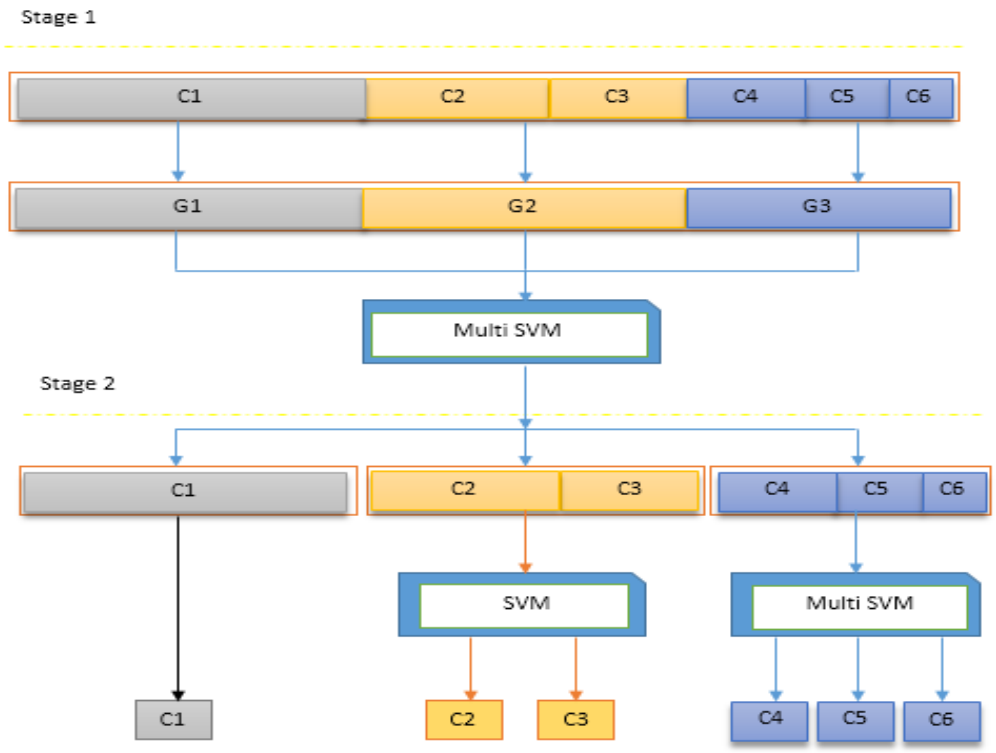


Fig. 3-2:How Does the proposed multi-stages model work?

3.2.3.1. **How does the model work?**

The model goes through two main stages:

STAGE ONE: Treat the Imbalance situation:

We decompose the classification stages into a series of sub-decisions stages. The dataset classes will be reorganized in new groups such that the differences between the number of the instances in the groups is almost or nearly balanced, regardless to the number of the classes in each group. So, a group may include just a class or more.

We achieve the previous step through Grouping algorithm that originates new artificial balanced groups. The algorithm goes through the following procedures:

1. Reorder the classes decently according to the number of the instances in each class, i.e. the classes' sizes {C_1, C_2, C_3…C_L}.
2. Starting from the last class C_L in the ordered list of the classes, add the number of its instances (#Ex$C_{L)}$ to those belong to the former classes in the ordered classes list {#ExC_L+#ExC_{L-1}+…. = SUM} till the accumulated summation becomes bigger than the number of the instances of class C_1 (the class at the top of the ordered list that contains the biggest number of instances).
3. If the difference between the accumulated summation (SUM) and the number of the instances of the class at the top of the ordered class (C_1) is less than the difference between the number of the instance of that corresponding class (C_N) and C_1 then join all the classes starting from the last class C_1 up to C_N in one group G_1 and the each one of the rest of the classes {C_{N-1}…C_1} will be in an independent group.
4. Start new level in the hierarchy.
5. Repeat the previous procedure to the classes in $G_{1,}$ noticing that the class C_N will be the top of its ordered classes. Then repeat them in every new formed group till regroup all the dataset classes following the same way.

STAGE TWO: The Mutli-stages of Classification:

After reorganizing the original dataset in new sub datasets, each one will be examined by an independent SVM machine.

At each level in the hierarchy, if the SVM decides to assign some tested example to an internal group that contains two classes or more, a new SVM will be applied to that group to assign the example for one of the classes it contains.

3.2.3.2. Classes Grouping Algorithm

For better explain the way the algorithm works, it was written in pseudo-code as Figure 3-3 illustrates.

Classes Grouping Algorithm

Input: n: Number of class; $x[n]$: Array of Number of samples for each class

Output: New balanced Groups

```
1: repeat
2:     Let j=0
3:     Let y[0] =x[0]
4:     repeat
5:         Let j=j+1
6:         Let y[j]=y[j-1] +x[j];
7:         Let t=j
8:     until y[j]<x[n-1]
9:     if ((y[t]-x[n-1])>(x[n-1]-x[t])) then
10:        return a new group including the considering class only and another
group contains the rest of the classes
11:    Let n=t
12:    else
13        return a new group including the considering class as well as the rest of
the classes
14:    n=t+1;
15:    end if
16: until t>1
```

Fig. 3-3:Classes Grouping Algorithm

The algorithm is built using C++ programming language, the output of the program will be the input of the classification process; the next part of the model.

3.2.4. Phase Four: Select Benchmark Datasets

This phase aims at electing Multiclass imbalanced datasets from U.C.I Repository from different fields, with distinctive characteristics to test the model.

3.2.5. Phase Five: Apply the model over the selected Benchmark datasets

Each dataset will be examined by four machines:

- SVM without weight
- SVM with weight
- The proposed model without weight.
- The proposed model with weight.

The previous two phases will be discussed in the next chapter in detail.

3.2.6. Phase Six: Performance Evaluation

The main objective of this phase is to identify evaluation criteria for the proposed model. This can be achieved through two steps: firstly, review the evaluation performance metrics for the performance of the classifiers of binary imbalanced data that can be extended to the multiclass situation as well as the metrics of hierarchical classification, to better choose the most suitable. This step will be done through two steps: firstly, reviewing the most important interesting metrics in chapter five. Secondly, investigating the overall performance of our model empirically in terms of some selected measures. This will be done in chapter six.

3.3. Summary

This chapter presented the research phases, how each phase is conducted, and how these phases are related. It detailed the design of the proposed classification model, as well as its techniques to treat the imbalance and Multiclass nature of the data.

4. CHAPTER FOUR: Experiments

4.1. Introduction:

This chapter describes the implementation phase of this research. This phase consists of selecting benchmark datasets from U.C.I Repository for setting up the experiments. They are chosen from different disciplines such as medicine, physics and biology, considering varieties in their properties to better test the model. Their description is detailed in this chapter as well as illustrating the first part of the proposed model, which is implementing the Grouping Algorithm to rebalance the data.

4.2. The Experiments Setup:

For the experimental setup, we ran 10 iterations of 10-fold cross-validation. Nine popular imbalanced data sets were selected from U.C.I. Repository. The original webpage where the data set can be found is: http://archive.ics.uci.edu/ml/datasets. The data sets are from different fields such as biology, physics, medicine, etc. While choosing these data sets, we tried to cover the range of variety in the data sets properties. The selection was based on:

- A range of Imbalance Ratio (IR) measure values.
- Variation in number of Classes (#Class),
- A varying number of total examples (#Examples) and number of attributes (#Attributes).

Each dataset will be examined by four machines:

- SVM without weight.
- SVM with weight.
- The proposed model without weight.

- The proposed model with weight.

The following table shows the selected benchmark datasets with their characteristics:

Table 4-1:The Benchmark Datasets & their Statistics

	Name	#Attributes	#Examples in each Class	#Total Examples	IR
1	**Yeast**	8	244/429/463/44/35/51/163/30/20/5	1484	23.15
2	**New-Thyroid**	5	150/35/30	215	4.84
3	**Dermatology**	34	112/61/72/52/49/20	366	5.55
4	**Balance**	4	49/288/288	625	5.88
5	**Glass Identification**	9	70/76/17/13/9/29	214	8.44
6	**Thyroid**	21	666/17/37	720	36.94
7	**Ecoli**	7	143/77/2/2/35/20/5/52	336	71.5
8	**Page Blocks**	10	492/33/12/8/3	548	164
9	**Shuttle**	9	1706/338/123/6/2	2175	853

4.3. Datasets Details Description:

The following tables present the different characteristics of the nine selected datasets:

- **Dataset** 1: **YEAST**

  It aims at predicting the Cellular Localization Sites of Proteins. Its source is Kenta Nakai, Institue of Molecular and Cellular Biology, Osaka University.

  Table 4-2 describes main characteristics of the **Yeast** dataset

Table 4-2: Yeast Imbalanced Multi-class data set

Type	**Imbalanced Multiclass**	**Origin**	**Real world**
Features	8	(Real / Integer / Nominal)	(8 / 0 / 0)
Instances	1484	IR	23.15
% Positive instances	4.14	% Negative instances	95.86
Missing values?	No		

Table 4-3 describes the attributes of the **Yeast** dataset

Table 4-3: Yeast Dataset Attributes

Attribute	**Domain**
mcg	[0.11, 1.0]
gvh	[0.13, 1.0]
alm	[0.21, 1.0]
mit	[0.0, 1.0]
erl	[0.5, 1.0]
pox	[0.0, 0.83]
vac	[0.0, 0.73]
nuc	[0.0, 1.0]
Class	{MIT, NUC, CYT, ME1, ME2, ME3, EXC, VAC, POX, ERL}

In order to describe the Grouping algorithm details, a number of abbreviations and colored cells are used. Table 4-4 illustrates the meaning of each:

Table 4-4: Abbreviations & colored cells

HS:	Highest number of sample

i:	The class number in the descendly ordered list
S(i):	Summation of the classes {C1, C2…Ci}
S(t):	Summation of the classes {C1, C2…Ct}
The yellow cell:	indicates to the biggest number of examples
The dark blue cell (t):	Indicates to the examples number of the corresponding class which will be tested either to be included alone in a group or to be joined to the rest of the classes in a group
The light blue cells	The Summation of the samples of the descendly ordered classes
The green cell:	Indicates that the corresponding class cell will be separated in a new group
The red cell:	Indicates that the corresponding class cell will be included with its following classes in a new group

Table 4-5, Table 4-6, Table 4-7, Table 4-8, and Table 4-9 clarify the steps of applying the Grouping Algorithm of the model over the YEAST dataset. It will be applied over the rest of the selected datasets in the same way.

Table 4-2:Classification of Applying the Grouping Algorithm over YEAST dataset classes: STEP1

	i	Class	Samples	Summation	HS-S(i)	HS-S(t)
	10	CYT	463	1484	-1021	
	9	NUC	429	1021	-558	
t	8	MIT	244	592	-129	219
	7	ME3	163	348	115	
	6	ME2	51	185	278	
	5	ME1	44	134	329	
	4	EXC	35	90	373	
	3	VAC	30	55	408	
	2	POX	20	25	438	

	1	ERL	5	5	458	

Table 4-3:Classification of Applying the Grouping Algorithm over YEAST dataset classes: STEP2

	i	Class	Samples	Summation	HS-S(i)	HS-S(t)
	8	MIT	244	592	-348	
t	7	ME3	163	348	-104	81
	6	ME2	51	185	59	
	5	ME1	44	134	110	
	4	EXC	35	90	154	
	3	VAC	30	55	189	
	2	POX	20	25	219	
	1	ERL	5	5	239	

Table 4-4:Classification of Applying the Grouping Algorithm over YEAST dataset classes: STEP3

	i	Class	Samples	Summation	HS-S(i)	HS-S(t)
	6	ME2	51	185	-134	
	5	ME1	44	134	-83	
	4	EXC	35	90	-39	
t	3	VAC	30	55	-4	21
	2	POX	20	25	26	
	1	ERL	5	5	46	

Table 4-5:Classification of Applying the Grouping Algorithm over YEAST dataset classes: STEP4

	i	Class	Samples	Summation	HS-S(i)	HS-S(t)
	3	VAC	30	55	-25	0
	2	POX	20	25	5	
	1	ERL	5	5	25	

Table 4-6:Classification of Applying the Grouping Algorithm over YEAST dataset classes: STEP5

	i	Class	Samples	Summation	HS-S(i)	HS-S(t)
	2	POX	20	25	-5	0
	1	ERL	5	5	15	

Figure 4-1 shows how the dataset 1 (**YEAST**) classes will be formed in multiple stages by the model**:**

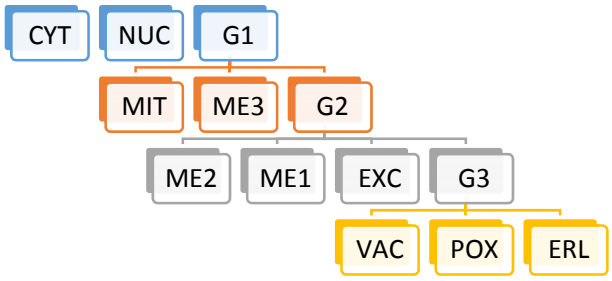


Fig. 4-1:Applying the Grouping Algorithm over Dataset 1

- **Dataset 2: New- Thyroid**

Thyroid Diseases is a medical condition harming the function of the thyroid. There are different thyroid diseases that have a broad range of symptoms and affect all ages. [92] Its source is Ross Quinlan From Garavan Institute.

Table 4-10 describes main characteristics of the **New-Thyroid** dataset:

Table 4-7: Thyroid Dataset Characteristics

Type	**Imbalanced Multiclass**	**Origin**	**Real world**

Features	5	(Real / Integer / Nominal)	(4 / 1 / 0)
Instances	215	IR	4.84
% Positive instances	17.12	% Negative instances	82.88
Missing values?	No		

Table 4-11 describes the **Thyroid Disease** dataset attributes

Table 4-8:Thyroid Dataset Attributes

Attribute	Domain
T3resin	[65, 144]
thyroxin	[0.5, 25.3]
triiodothyronine	[0.2, 10.0]
thyroidstimulating	[0.1, 56.4]
TSH_value	[-0.7, 56.3]
class	{normal, hyper, hypo}

Figure 4-2 shows how the dataset 2 classes will be formed in multiple stages by the model:

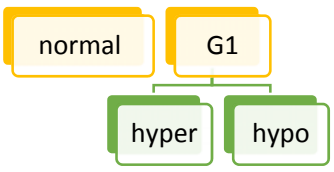


Fig. 4-2:Applying the Grouping Algorithm over Dataset 2

- **Data set 3: Dermatology**

Dermatology is the branch of medicine dealing with the skin, nails, hair and its diseases. It is a specialty with both medical and surgical aspects. [93] Its aim is at determining the type of Eryhemato-Squamous Disease. Table 4-12 describes main characteristics of the **Dermatology** dataset:

Table 4-9:Dermatology Dataset Characteristics

Type	**Imbalanced Multiclass**	**Origin**	**Real world**
Features	34	(Real / Integer / Nominal)	(0 / 34 / 0)
Instances	366	IR	5.55
% Positive instances	15.27	% Negative instances	84.73
Missing values?	Yes		

Table 4-13 describes the **Dermatology** dataset attributes**:**

Table 4-10:Dermatology Dataset Attributes

Attribute	**Domain**	**Attribute**	**Domain**	**Attribute**	**Domain**
a1	[0,3]	a13	[0,2]	a24	[0,3]
a2	[0,3]	a14	[0,3]	a25	[0,3]
a3	[0,3]	a15	[0,3]	a26	[0,3]
a4	[0,3]	a16	[0,3]	a27	[0,3]
a5	[0,3]	a17	[0,3]	a28	[0,3]
a6	[0,3]	a18	[0,3]	a29	[0,3]
a7	[0,3]	a19	[0,3]	a30	[0,3]
a8	[0,3]	a20	[0,3]	a31	[0,3]
a9	[0,3]	a21	[0,3]	a32	[0,3]
a10	[0,3]	a22	[0,3]	a33	[0,3]
a11	[0,1]	a23	[0,3]	a34	[0,75]

a12	[0,3]	class	[1,6]		

Figure 4-3 shows how the dataset 3 classes will be formed in multiple stages by the model

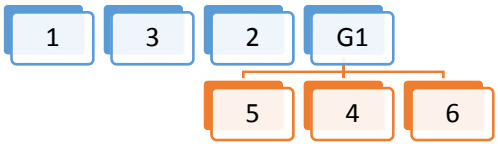


Fig. 4-3:Applying the Grouping Algorithm over Dataset 3

- **Dataset 4: Balance Scale Dataset**

This data set was generated to model psychological experimental results by Siegler, R. S. (1976). Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced.

Table 4-14 describes main characteristics of the **Balance** dataset:

Table 4-11:Balance Scale Dataset Characteristics

Type	**Imbalanced Multiclass**	**Origin**	**Real world**
Features	4	(Real / Integer / Nominal)	(4 / 0 / 0)
Instances	625	IR	5.88
% Positive instances	14.53	% Negative instances	85.47
Missing values?	No		

Table 4-15 describes the **Balance** data set attributes:

Table 4-12:Balance Scale Dataset attributes

Attribute	Domain
left-weight	[1.0, 5.0]
left-distance	[1.0, 5.0]
right-weight	[1.0, 5.0]
right-distance	[1.0, 5.0]
class	{L, B, R}

When applying the Grouping algorithm over the Balance dataset whose classes contain these number of examples: 4, 288,288, the algorithm decides to include the third class as a one group as well as each one of the other classes according to its measure in forming the new groups that depends on the least difference between the classes sizes. So, it changes nothing in the class and their instances distribution. In other words, the new formed groups are identical to the original classes. Therefore, the model is not applicable considering this dataset.

- **Dataset 5: Glass Identification**

From USA Forensic Science Service, six types of glass have been defined in terms of their oxide content (i.e. Na, Fe, K, etc). Its source is B. German, Central Research Establishment.

The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence...if it is correctly identified.

Table 4-16 describes main characteristics of the **Glass Identification** data set:

Table 4-13:Glass Identification Dataset Characteristics

Type	Imbalanced Multiclass	Origin	Real world
Features	9	(Real / Integer / Nominal)	(9 / 0 / 0)
Instances	214	IR	8.44
% Positive instances	10.59	% Negative instances	89.41
Missing values?	No		

Table 4-17 describes the **Glass Identification** data set attributes:

Table 4-14:Glass Identification Dataset Attributes

Attribute	Domain
RI	[1.51115, 1.53393]
Na	[10.73, 17.38]
Mg	[0.0, 4.49]
Al	[0.29, 3.5]
Si	[69.81, 75.41]
K	[0.0, 6.21]
Ca	[5.43, 16.19]
Ba	[0.0, 3.15]
Fe	[0.0, 0.51]
type Glass	{1, 2, 3, 4, 5, 6, 7}

Figure 4-4 shows how the dataset 5 classes will be formed in multiple stages by the model:

Fig. 4-4:Applying the Grouping Algorithm over Dataset 5

- **Dataset 6: Thyroid Disease (Thyroid0387)**

Table 4-18 describes main characteristics of the **Thyroid** dataset:

Table 4-15:Thyroid Disease (thyroid0387) Dataset Attributes

Type	**Imbalanced Multiclass**	**Origin**	**Real world**
Features	21	(Real / Integer / Nominal)	(6 / 0 / 15)
Instances	720	IR	36.94
% Positive instances	2.64	% Negative instances	97.36
Missing values?	No		

Table 4-19 describes the attributes of the **Thyroid** dataset:

Table 4-16:Thyroid Disease (thyroid0387) Dataset Attributes

Attribute	**Domain**	**Attribute**	**Domain**	**Attribute**	**Domain**
Sintoma1	[0.01, 0.97]	Sintoma8	[0, 1]	Sintoma15	[0, 1]
Sintoma2	[0, 1]	Sintoma9	[0, 1]	Sintoma16	[0, 1]
Sintoma3	[0, 1]	Sintoma10	[0, 1]	Sintoma17	[0.0, 0.53]
Sintoma4	[0, 1]	Sintoma11	[0, 1]	Sintoma18	[0.0005, 0.18]
Sintoma5	[0, 1]	Sintoma12	[0, 1]	Sintoma19	[0.0020, 0.6]
Sintoma6	[0, 1]	Sintoma13	[0, 1]	Sintoma20	[0.017, 0.233]

Sintoma7	[0, 1]	Sintoma14	[0, 1]	Sintoma21	[0.0020, 0.642]
class	{1,2,3}				

Figure 4-5 shows how the dataset 6 classes will be formed in multiple stages by the model

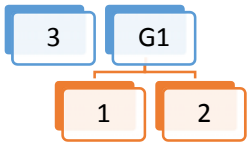


Fig. 4-5:Applying the Grouping Algorithm over Dataset 6

- **Dataset 7: Ecoli Imbalanced Multi-class dataset**

This data contains protein localization sites. Its source is Kenta Nakai , Institue of Molecular and Cellular Biology, Osaka, University.

Table 4-20 describes main characteristics of the **Ecoli** dataset:

Table 4-17:Ecoli Dataset Attributes

Type	**Imbalanced Multiclass**	**Origin**	**Real world**
Features	7	(Real / Integer / Nominal)	(7 / 0 / 0)
Instances	336	IR	71.5
% Positive instances	1.38	% Negative instances	98.62
Missing values?	No		

Table 4-21 describes attributes of the **Ecoli** dataset:

Table 4-18:Ecoli Dataset Attributes

Attribute	Domain
mcg	[0.0, 0.89]
gvh	[0.16, 1.0]
lip	[0.48, 1.0]
chg	[0.5, 1.0]
aac	[0.0, 0.88]
alm1	[0.03, 1.0]
alm2	[0.0, 0.99]
class	{cp, im, pp, imU, om, omL, imL, imS}

Figure 4-6 shows how the dataset7 classes will be formed in multiple stages by the model

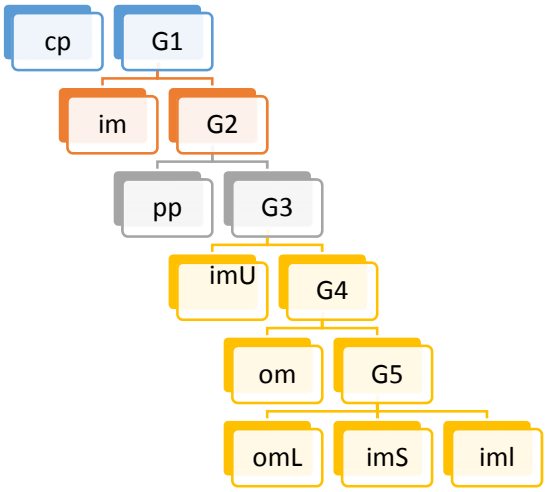


Fig. 4-6:Applying the Grouping Algorithm over Dataset 7

- **Dataset 8: Page Blocks Imbalanced Multi-class dataset**

The problem consists of classifying all the blocks of the page layout of a document that has been detected by a segmentation process. The 5473 examples come from 54 distinct documents. Each observation concerns one

block. Its source is Donato Malerba, Dipartimento di Informatica, University of Bari.

Table 4-23 describes main characteristics of the **Page blocks** dataset:

Table 4-19:Page Blocks Dataset Characteristics

Type	Imbalanced	Origin	Real world
Features	10	(Real / Integer / Nominal)	(10 / 0 / 0)
Instances	548	IR	164
% Positive instances	0.61	% Negative instances	99.39
Missing values?	No		

Table 4-24 describes attributes of the **Page Blocks** dataset:

Table 4-20:Page Blocks Dataset Attributes

Attribute	Domain
height	[1.0, 804.0]
lenght	[1.0, 553.0]
area	[7.0, 143993.0]
eccen	[0.0070, 537.0]
p_black	[0.052, 1.0]
p_and	[0.062, 1.0]
mean_tr	[1.0, 4955.0]
blackpix	[1.0, 33017.0]
blackand	[7.0, 46133.0]
wb_trans	[1.0, 3212.0]
class	{1, 2, 4, 5, 3}

Figure 4-7 shows how the dataset 8 classes will be formed in multiple stages by the model:

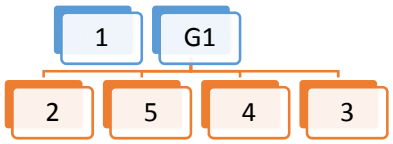


Fig. 4-7:Applying the Grouping Algorithm over Dataset 8

- **Dataset 9: Statlog (Shuttle) Imbalanced Multi-class Dataset**

Approximately 80% of this dataset belongs to class1. Table 4-25 describes the main characteristics of the **Shuttle** dataset:

Table 4-21:Statlog (Shuttle) Dataset Characteristics

Type	**Imbalanced Multiclass**	**Origin**	**Real world**
Features	9	(Real / Integer / Nominal)	(0 / 9 / 0)
Instances	2175	IR	853
% Positive instances	0.12	% Negative instances	99.88
Missing values?	No		

Table 4-26 describes the attributes of the **Shuttle**

Table 4-22:Statlog (Shuttle) Dataset Attributes

Attribute	**Domain**
a1	[27, 126]
a2	[-4821, 5075]
a3	[21, 149]
a4	[-3939, 3830]
a5	[-188, 436]
a6	[-13839, 13148]
a7	[-48, 105]
a8	[-353, 270]

a9	[-356, 266]
class	{1,2,3,4,5,6,7}

Figure 4-8 shows how the dataset 9 classes will be formed in multiple stages by the model:

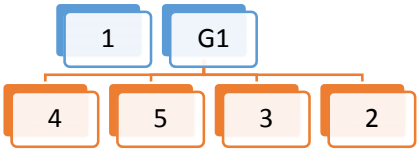


Fig. 4-8:Applying the Grouping Algorithm over Dataset 9

After applying the grouping algorithm – the first part of the model to rebalance the data, the output will be the input of the Support Vector Machine or/and Multi-Support Vector Machine utilizing the environment of the RapidMiner 5.3.007 software package. It is open-source Java-based DM software. It is free software. It can be downloaded and installed from RapidMiner home page http://rapid-i.com.

For each one of the considering classification method the same training, validation and testing sets were used. Initially, a brief identification of X-Validation method is introduced:

4.4. X-Validation

The X-Validation is Rapid Miner name for the K-fold cross-validation. To estimate the statistical performance of a learning operator, the X-Validation operator in RapidMiner executes a cross-validation. This operator partitions the input dataset into k subsets of equal size. From the k subsets, a single subset is retained as the testing dataset (i.e. input of the testing), and the remaining $k - 1$ subsets are used as training dataset. The cross-validation

process is then repeated ***k*** times, with each of the ***k*** subsets used exactly once as the testing data. The ***k*** results from the ***k*** iterations can then be averaged (or otherwise combined) to produce a single estimation. The value ***k*** was chosen to be 10 in these experiments

4.5. Sampling type

RapidMiner provides several types of sampling for building the training and testing subsets.

- **Linear sampling**: It simply splits the dataset into partitions (training and testing according to specified splitting ratio) without changing the order of the instances (tuples), this means; many subsets with consecutive instances are produced.
- **Shuffled sampling**: It builds random subsets of a dataset. Instances are elected randomly for building subsets.
- **Stratified sampling**: Stratified sampling forms random subsets and confirms that the class distribution in the subsets is the same as in the whole dataset. For example, in the case of a binominal classification, stratified sampling forms random subsets such that each subset includes roughly the same proportions of the two values of class labels.

In all experiments, the last types were used due to its suitability to the chosen datasets.

4.6. Summary:

This chapter illustrated applying the suggested hierarchical model and explored each dataset characteristics.

5. CHAPTER FIVE: Performance Evaluation Metrics

5.1. Introduction

This chapter highlights the evaluation metrics for binary imbalanced classifiers which was extended to assess the performance of the classifiers of the multiclass and multiclass imbalanced data. It details the first type of these metrics since it's the most used, so we have considered it in this study for the performance evaluation process.

5.2. Evaluation Metrics for Binary Imbalanced Data:

There are three families of evaluation metrics used in the context of classification [94], [95]. These are: **The Threshold Metrics** (e.g., accuracy and F-measure), **The Ranking Metrics** (e.g., Receiver Operating Characteristics (ROC) analysis and AUC), and **The Probabilistic Metrics** (e.g., Root-Mean-Squared Error (RMSE)). When handling imbalanced learning problems, traditional evaluation techniques may not be capable of providing a sensible and comprehensive assessment of the imbalanced learning algorithms. Studies showed that an individual evaluation metric, such as overall classification error rate, and overall accuracy are not satisfactory. A combination of threshold metrics (e.g., precision, recall, F-measure, and G-mean) together with ranking assessment metrics [e.g., receiver operating characteristic (ROC) curve, precision–recall (PR) curve, and cost curve) will achieve more complete assessment of imbalanced learning. A review of the main evaluation metrics and their advantages and disadvantages with respect to the class imbalance problem will be highlighted in this chapter.

5.2.1. THRESHOLD METRICS: Singular Assessment Metrics

Threshold metrics can have a multiple - or a single - class focus [96]. The multiple-class focus metrics consider the overall performance of the learning algorithm on all the classes in the dataset. Some take class ratios into consideration of them like accuracy, error rate which will be discussed in this subsection as well as the single - class focus measures such as sensitivity/specificity, precision/recall, Geometric mean (G-mean), and F-measure. All the metrics discussed in this section are based on the concept of the confusion matrix.

Considering a basic two-class classification problem, let {p, n} be the true positive and negative class label and {Y, N} be the predicted positive and negative class labels. Then, a representation of classification performance can be formulated by a **Confusion Matrix** (contingency table), as illustrated in Figure 5-1.

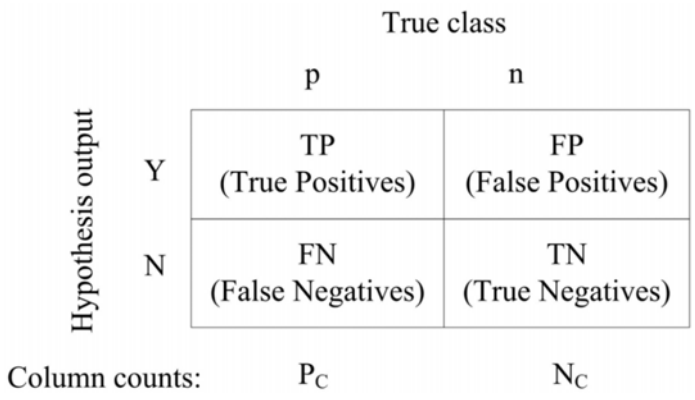


Fig. 5-1:Confusion Matrix for Performance Evaluation for two classes

The true positive and true negative entries indicate the number of examples correctly classified by classifier $\boldsymbol{f}$ as positive and negative, respectively. The false negative entry indicates the number of positive examples wrongly classified as negative. Conversely, the false positive entry indicates the

number of negative examples wrongly classified as positive. If we consider the minority class as the positive class and the majority class as the negative class. Accuracy and Error Rate are defined as:

$$\textbf{Accuracy} = \frac{\textbf{TP} + \textbf{TN}}{\textbf{PC} + \textbf{NC}} \quad ; \quad \text{Error Rate} = 1 - \text{Accuracy} \ldots\ldots \quad (5.1)$$

These metrics provide a straightforward way of describing a classifier's performance on a given data set. As previously mentioned, they are highly sensitive to changes in data, but they can be misleading in such a situation where a given data set includes five percent of minority class examples and 95 percent of majority examples, then classifying every example to be a majority class example would provide an accuracy of 95 percent which is very great when assessing the classifier, but this assessment fails to reflect the fact that 0 percent of minority examples are identified. Many studies had agreed with the ineffectiveness of accuracy in the imbalanced learning scenario [14], [97], [98], [99] , so, other evaluation metrics were adopted to provide assessments of imbalanced binary learning problems. These metrics are defined as:

$$\textbf{Sensitivity} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}} \quad (5.2)$$

$$\textbf{Specificity} = \frac{\textbf{TN}}{\textbf{FP} + \textbf{TN}} \quad (5.3)$$

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}} \quad (5.4)$$

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}} \quad (5.5)$$

$$\textbf{F-measure} = \frac{(\mathbf{1} + \boldsymbol{\beta})^{\mathbf{2}} . \textbf{Recall} . \textbf{Precision}}{\boldsymbol{\beta} . \textbf{Recall} . \textbf{Precision}} \quad (5.6)$$

where β is a coefficient to adjust the relative importance of precision versus recall (usually $\beta = 1$).

$$\textbf{G-mean} = \sqrt{\frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}} \times \frac{\textbf{TN}}{\textbf{TN} + \textbf{FP}}} \quad (5.7.1)$$

OR

$$\textbf{G-mean} = \sqrt{\textbf{Sensitivity} \times \textbf{Specificity}}$$

$$\textbf{G-mean} = \sqrt{\textbf{Sensitivity} \times \textbf{Precision}} \quad (5.7.2)$$

$$\textbf{MMA} = \frac{\textbf{Sensitivity} + \textbf{Specificity}}{\textbf{2}} \quad (5.8)$$

$$\textbf{MCWA} = w \times \text{sensitivity} + (1 - w) \times \text{specificity} \quad (5.9)$$

where **w** is a value between 0 and 1, which represents the weight assigned to the positive class.

$$\textbf{AGm} = \frac{\textbf{G} - \textbf{mean} + \textbf{specificity} \times \textbf{Nn}}{\textbf{1} + \textbf{Nn}}, \text{ if Specificity} > 0 \quad (5.10.1)$$

$$\textbf{AGm} = 0, \text{ if Sensitivity} = 0 \quad (5.10.2)$$

$$\textbf{Optimized Precision} = \textbf{Specificity} \times \textbf{Nn} + \textbf{Sensitivity} \times \textbf{Np} - \frac{|\textbf{Specificity} - \textbf{Sensitivity}|}{\textbf{Specificity} + \textbf{Sensitivity}} \quad (5.11)$$

where ***Nn*** represents the number of negative examples in the dataset, and ***Np*** represents the number of positive examples in the dataset.

$$\text{IBA}\alpha(M) = (1 + \alpha \times \text{Dom}) \times M \quad (5.12)$$

where dominance ***(Dom)*** is defined as: ***Sensitivity – Specificity***, M is any metric, and ***α*** is a weighting factor designed to reduce the influence of the dominance on the result of a particular metric ***M***.

- **Sensitivity and Specificity:**

The Sensitivity of a classifier ***f*** refer to its true positive rate or the proportion of positive examples actually assigned as positive by ***f***, while the complement metric to it is called the Specificity of classifier ***f***. It corresponds to the proportion of negative examples that are discovered. It is the same quantity, only it is measured over the negative class. They are defined as equation (5.2). They identify together the proportions of the two classes correctly classified, but separately in the context of each individual class of instances unlike accuracy, so, the class imbalance does not affect these measures. Also, the cost of using these metrics appears in the form of a metric for each single class, which is more difficult to process than a single measure. This pair of metrics misses the measure of the proportion of examples assigned to a given class by classifier ***f*** that actually belongs to this class. Instead they, together, identify the proportions of the two classes correctly classified.

- **Precision & Recall:**

The precision of a classifier ***f*** measures how precise ***f*** is when identifying the examples of a class, i.e. it assesses the proportion of examples assigned a positive classification that are truly positive. This quantity together with sensitivity, which is commonly called Recall when considered together with precision, is typically used in the information retrieval context where researchers are interested in the proportion of relevant information identified

along with the amount of actually relevant information from the information assessed as relevant by ***f***. [1]

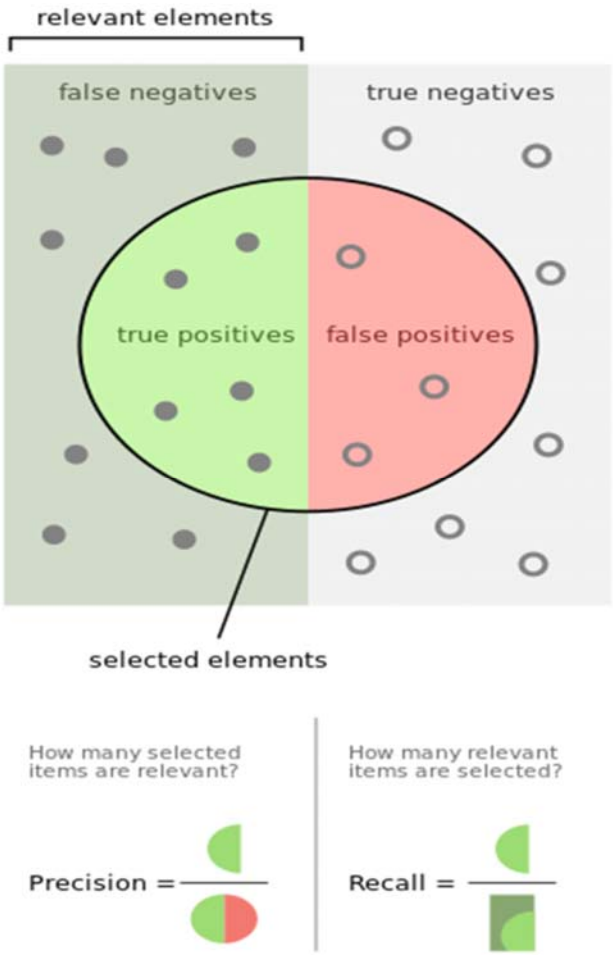


Fig. 5-2:Precision & Recall

In a classification task, Precision (5.4) for a class is a measure of exactness or true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class), whereas Recall (5.5) in this context is defined as a measure of completeness (i.e. the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and

false negatives, Figure 5-2 (which are items which were not labeled as belonging to the positive class but should have been)). These two metrics, share an inverse relationship between each other much like accuracy and error rate, but Precision is sensitive to data distributions while Recall is not. Since Recall provides no insight to how many examples are incorrectly labeled as positive, it can be equivocal evaluation if consider it solely. Similarly, Precision cannot assert how many positive examples are labeled incorrectly. The focus is on the positive class only, meaning that the problems encountered by multi-class focus metrics in the case of the class imbalance problem are, once more, avoided. As for sensitivity and specificity, however, the cost of using precision and recall is that two measures must be considered and that absolutely no information is given on the performance of f on the negative class. This information did appear in the form of specificity in the previous pair of metrics. Nevertheless, they can effectively evaluate classification performance in imbalanced learning scenarios.

- **The F-Measure:**

It combines precision and recall as a measure of the effectiveness of classification in terms of a ratio of the weighted importance on either recall or precision as determined by the β coefficient that set by the user. However, being sensitive to data distributions, F-Measure provides deep view into the functionality of a classifier than the accuracy metric. Its formulation was defined in equation (5.6).

- **Geometric Mean (G-Mean):**

It was introduced by Kubat et al. [100] as a response to the class imbalance problem and as a response to the fact that a single metric is easier

to deploy than a pair of metrics. It considers the relative balance of the classifier's performance on both the positive and the negative classes. So, it is defined as a function of both the sensitivity and the specificity of the classifier as in equation (5.1). While being more sensitive to class imbalances than accuracy - because the two classes are given equal importance -, it remains close to the multi-class focus category of metrics. So, another version of the G-mean was introduced to focus solely on the positive class. It replaces the specificity term by the precision term, yielding equation (5.2) and evaluates the degree of inductive bias in terms of a ratio of positive accuracy and negative accuracy. Though, F-Measure and G-Mean are great improvements over accuracy, they are still ineffective in answering more generic questions about classification evaluations, like how can we compare the performance of different classifiers over a range of sample distributions?

An important disadvantage of all the threshold metrics is that they assume full knowledge of the conditions under which the classifier will be deployed. Particularly, they assume that the class imbalance present in the training set is the one that will be encountered throughout the operating life of the classifier. If that is truly the case, then the previously discussed metrics are appropriate; however, it has been suggested that information related to skew (as well as cost and other prior probabilities) of the data is generally not known. In such cases, it is more useful to use evaluation methods that enable visualization or summarization performance over the full operating range of the classifier. Particularly, such methods perform the assessment of a classifier's performance over all possible imbalance or cost ratios, For instance, ranking methods: ROC curves, cost curves, precision–recall (PR)

curves, AUC or AUROC (Area under the ROC curve). But they are far from the scope of this study since we consider the extensions of the previous discussed metrics that can evaluate the performance of classifiers of the multiclass imbalanced data. [1]

5.3. Evaluation Metrics for Multiclass & Multiclass Imbalanced Data:

Many performance evaluation metrics of binary classifiers have been extended to suit the multiclass and they also utilized for imbalanced multiclass situation. They include all previously mentioned ones as well as those dedicated for hierarchical classification. These metrics can be classified into: distance-based, depth-dependent, semantics-based and hierarchy-based [101]. In addition, there are Multi-criteria Measures, such as interestingness and comprehensibility [96].

The threshold-metrics frontia based on the concept of the **Confusion Matrix** which extended for multi-class data as Figure 5-3 illustrates.

Class	0	1	2	…	j
0	TP	FN	FN	FN	FN
1	FP	TN	FN	FN	FN
2	FP	FN	TN	FN	FN
:	FP	FN	FN	TN	FN
j	FP	FN	FN	FN	TN

Fig. 5-3 :Confusion Matrix for Multi-class

As same as the binary situation, many other metrics were concluded basing on the extended Multiclass confusion metric:

Sensitivity (True Positive Rate) or **Recall** of minority class is known as the ratio of correctly classified examples from the minority class, meanwhile **Specificity** is known as the ratio of correctly excluded examples from the majority classes).

Mosley et al. [102] designed new performance measure specifically for model validation in the presence of multi-class imbalance that called **Class Balance Accuracy** or **Recall (j)** or **Acc (j)**. It was defined as:

For any C^k confusion matrix:

$$CBA = \frac{\sum_i^k \frac{c_{ii}}{\max(c_{i.}, c_{i.})}}{k} \qquad (5.13)$$

Where C^k denote a k×k confusion matrix or contingency table of actual class labels aligned by their model predictions, with cij representing the number of cases with true label i classified into group j and $c_{i.} = \sum_{j=1}^{k} c_{ij}$.

G-mean adapted by Sun & Kamel et al [103] to multi-class scenarios. It is defined as the geometric mean of the Recall values of all classes. Given a j-class problem:

$$\boldsymbol{G - mean} = \left(\prod_{i=1}^{j} Acc(i) \right)^{1/j} \qquad (5.14)$$

or

$$\mathbf{G - mean} = \frac{\sum_{i=1}^{J} Acc(i)}{\mathrm{J}} \qquad (5.14)$$

It can offer the balanced performance among minority and majority classes effectively, as the recognition rate of every class or the accuracies are balanced.

Considering cost-sensitive learning, it is natural to utilize misclassification costs for performance evaluation for multiclass imbalanced problems [104], [43], [70].

For the evaluation of learning algorithms based on class decomposition, some works chose to take the average of any two-class performance measure for produced binary classifiers [105], [86], [2].

Mean F−measure (MFM): this measure aggregates both the Precision and the Recall of the minority class. So, it can be illustrated as the weighted average of the Precision and Recall [71].

$$\mathbf{F-measure(j)} = \frac{2.Recall(j).Precision(j)}{Recall(j) + Precision(j)} \quad (5.15)$$

$$\mathbf{MFM} = \frac{\sum_{j=1}^{K} F - measure(j)}{K} \quad (5.16)$$

Kappa Statistic: It is a measure that compares the accuracy of the system to the accuracy of a random system [106].

$$\mathbf{Kappa} = \frac{\text{Total Accuracy} - \text{RandomAccuracy}}{1 - \text{RandomAccuracy}} \quad (5.17)$$

Total accuracy is simply the sum of true positive and true negatives, divided by the total number of items.

$$\textbf{Total Accuracy} = \frac{\sum TP + \sum TN}{Total\ of\ instances} \quad (5.18)$$

where

Random Accuracy is defined as the sum of the products of reference likelihood and result likelihood for each class. That is,

$$\textbf{RandomAccuracy} = \frac{(TN + FP) * (TN + FN) + (FN + TP) * (FP + TP)}{Total * Total} \quad (5.19)$$

Considering **Ranking Methods** for evaluation the scoring classifiers, **Multi-class ROC graphs** was proposed by Thomas and Robert et al. [71], it generates as many ROC curves as there are classes, where ROC curve originally is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is the sensitivity or recall. The false-positive rate is (1 - specificity) [23]. But the multiclass ROC graphs are sensitive to the class skew according to T. Fawcett et al. [107] and [108]. So, a pairwise approach is utilized by discounting some interactions, it approximates the multidimensional operating characteristic to obtain a tractable algorithm and can be extended to large numbers of classes to produce the multiclass ROC by pairwise analysis [71]. A ROC surface is defined for the Q-class problem as well, in terms of a multi-objective optimization problem utilizing evolutionary algorithm [109].

Another Ranking measure is **Multi-class AUC** which has been proposed to compute the weighted average of all the AUCs produced by the Multi-class ROC graph and a skew-sensitive version of this Multi-class AUC [70] where the Area Under the Curve is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. But under the multiclass imbalanced learning scenario, the AUC values for two-class problems become multiple pairwise discriminability values [110]. To calculate such multiclass AUCs, a probability estimation-based approach: First, the ROC curve for each reference class w_i is generated and their respective AUCs are measured. Second, all of the AUCs are combined by a weight coefficient according to the reference class's prevalence in the data. It was also sensitive to the class [111]. Moreover, M-measure or (MAUC) is a generalization approach that aggregates all pairs of classes based on the inherent characteristics of the AUC [23]. It is the average of AUC of all pairs of classes, and defined as:

$$M = \frac{2}{c(c-1)} \sum_{i<j} A(i,j) \qquad (5.20)$$

Where A (i, j) = [A (i|j) + A (j|i)] =2 for class pair (i, j). A (i, j) measures the separability between classes. A (i|j) is the probability that a randomly drawn example of class j will have a lower estimated probability of belonging to class i than a randomly drawn example of class i. It should be noted that AUC = A (i|j) = A (j|i) in the two-class scenario, but the equality does not hold when more than two classes exist. Another extension of the AUC measure to the multiclass case tended the volume under the ROC hypersurface that evaluates the VUS over the C-dimensional ROC surface [112].

The third sort of evaluation metrics used with the Probabilistic Classifiers, such as **RMSE** or **RMSD** which is used to measure the differences between values (sample and population values) predicted by the classifier and the values actually observed or estimated [113]. The RMSD of predicted values $\hat{y}_t$ for times t of the variable y_t is computed for n different predictions [114]:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}. \qquad (5.21)$$

Additionally, **Cosine Similarity** measures the similarity between two output categories as well as using **The Ranking Loss** which tends the order of the predicted score among C categories. They can also be deployed for probabilistic performance evaluation for multiclass [113]. A Bayesian framework is proposed for inferring on the posterior **balanced accuracy** [115]. The Balanced accuracy, i.e., by the arithmetic mean of class-specific accuracies is given by:

$$1/l \,.\, \sum_{i=1}^{l} \theta \mathrm{i} \qquad (5.22)$$

Where θi is the (latent) accuracy of the classifier on class i.

5.4. Evaluation Metrics for Hierarchies of Multiclass Imbalanced Data:

The hierarchical F-measure is a popular performance measure in hierarchical classification. It is defined as,

$$hF = \frac{2 * hP * hR}{hP + hR} \qquad (5.23)$$

$$hP = \frac{\sum_i \left|\widehat{P}_i \cap \widehat{T}_i\right|}{\sum_i \left|\widehat{P}_i\right|} \qquad (5.24)$$

$$hR = \frac{\sum_i \left|\widehat{P}_i \cap \widehat{T}_i\right|}{\sum_i \left|\widehat{T}_i\right|} \qquad (5.25)$$

Where *hP* is the hierarchical precision and *hR* is the hierarchical recall. ˆPi is the hierarchical categories predicted for test example *xi* while $T\hat{}_i$ is the true categories of x_i. This way to calculate the hierarchical was presented by Ipeirotis et al. [116] who utilized the concept of descendant classes in their performance evaluation by considering the subtrees rooted in the predicted class and in the true class. Each subtree is formed by the class itself and its descendants. The intersection of these subtrees is then used to calculate extended precision and recall measures [117].

To calculate the precision, the number of classes belonging to the intersection is divided by the number of classes belonging to the subtree rooted at the predicted class as defined by equation (5.24).

To calculate the recall, the number of classes belonging to the intersection is divided by the number of classes in the subtree rooted at the true class as defined by equation (5.25). To calculate a hierarchical extension of the F-measure, the hierarchical prediction and recall measures have to be obtained firstly [117].

The problem with this measure is that it assumes that the predicted class is either a subclass or a superclass of the true class. When these classes are in the same level, for example, their intersection is an empty set [117].

There are many other hierarchical performance metrics that differ in their way of work, but we concentrate on those used in this study.

5.5. Summary:

This chapter reviewed different metrics for evaluation the performance of binary imbalanced classifiers and their extensions to adopt assessing the performance of classifiers of imbalanced multiclass problems.

6. CHARTER SIX: Results Discussion

6.1. Introduction:

This chapter presents the conclusive results of the learning from Imbalanced Multiclass data. The results of each experiment will be illustrated in terms of the RECALL and the PRECISION and F-measure for each dataset, in addition to concluding remarks.

A comparison of the four classification machines in terms of OVERALL ACCURACY, G-mean, MFM and Kappa are also presented. As the result of these comparisons best technique(s) is (are) identified for each dataset. Finally, the conclusions of the all experimental results for this research are illustrated.

6.2. The elected Dataset.

- **Yeast Imbalanced Multiclass Dataset**

Figure 6-1, Figure 6-2 and Figure 6-3 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for Yeast dataset respectively.

Fig. 6-1:RECALL of Yeast Dataset

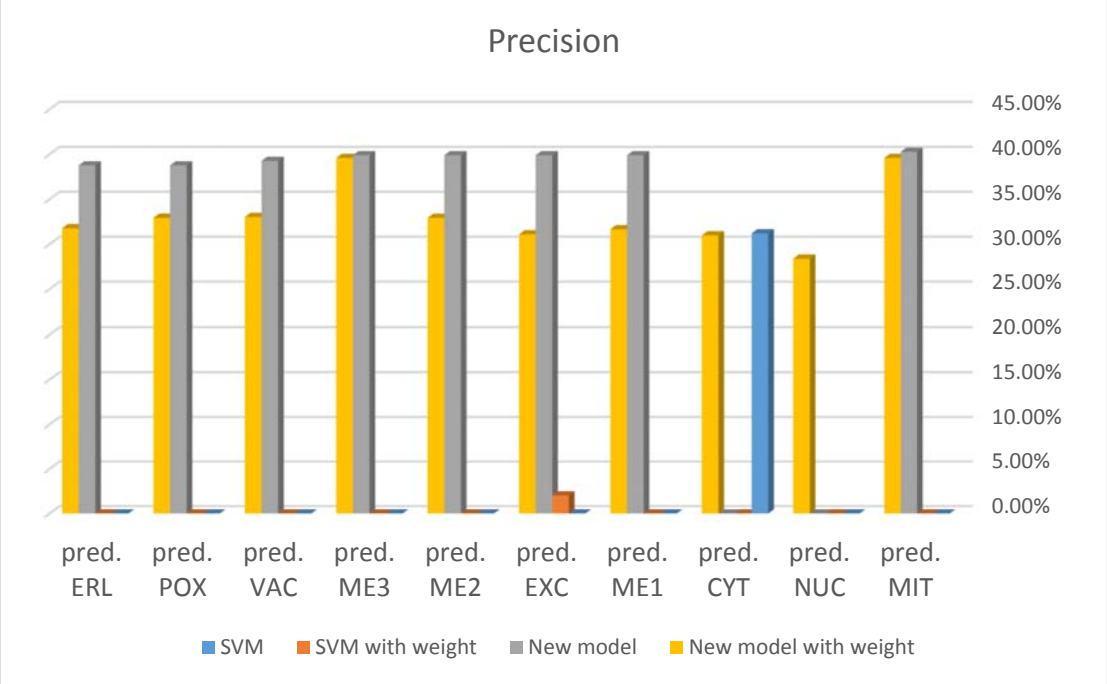


Fig. 6-2:PRECISION of Yeast Dataset

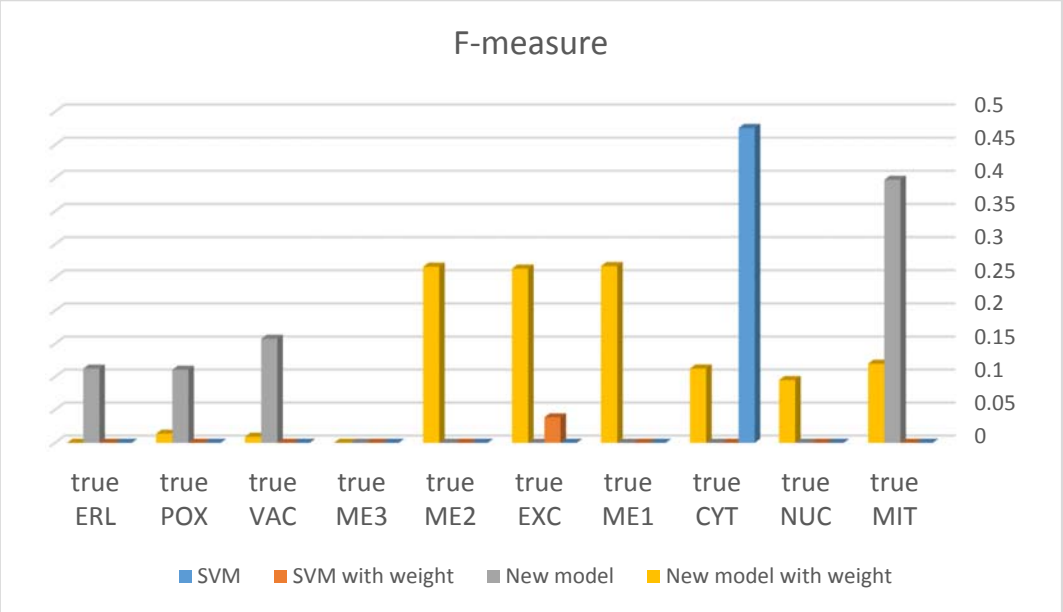


Fig. 6-3:F-measure of Yeast Dataset

- **New-Thyroid Imbalanced Multiclass Dataset**

Figure 6-4, Figure 6-5 and Figure 6-6 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for New-Thyroid dataset respectively.

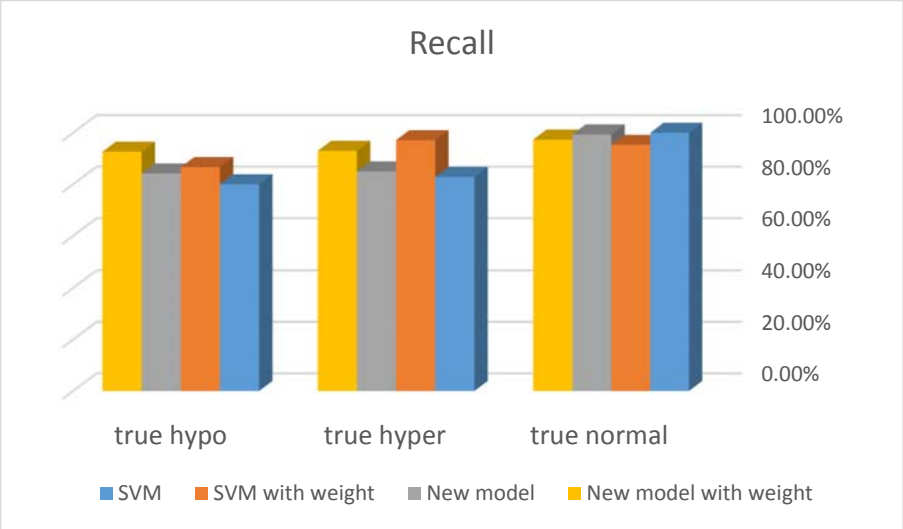


Fig. 6-4:RECALL of NEW-Thyroid Dataset

Fig. 6-5:PRECISION of NEW-Thyroid Dataset

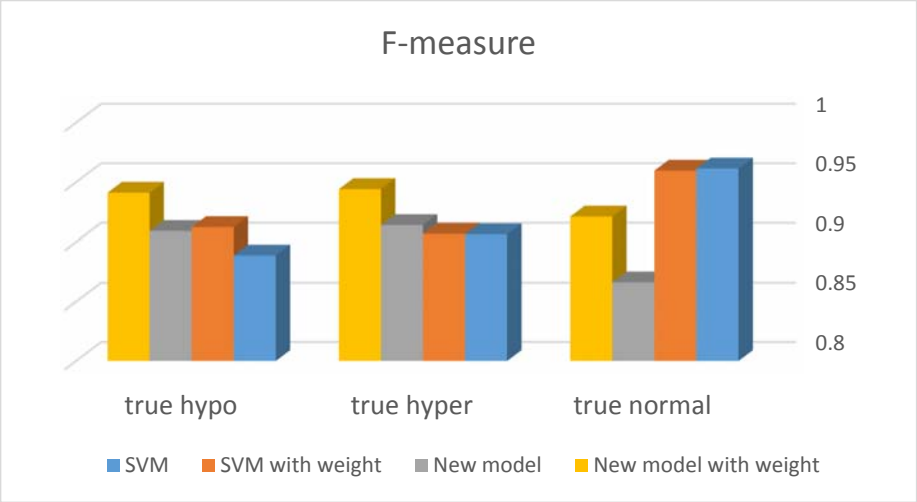


Fig. 6-6:F-measure of NEW-Thyroid Disease Dataset

- **Dermatology Imbalanced Multiclass Dataset**

Figure 6-7, Figure 6-8 and Figure 6-9 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for **Dermatology** dataset respectively.

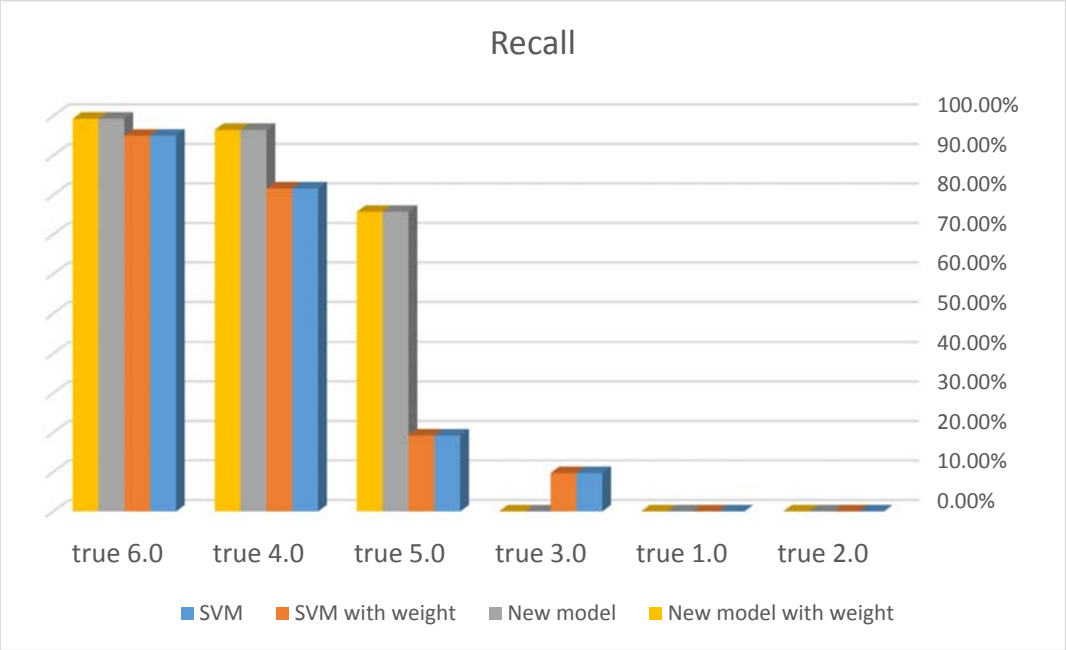


Fig. 6-7:RECALL of Dermatology Dataset

Fig. 6-8:PRECISION of Dermatology Dataset

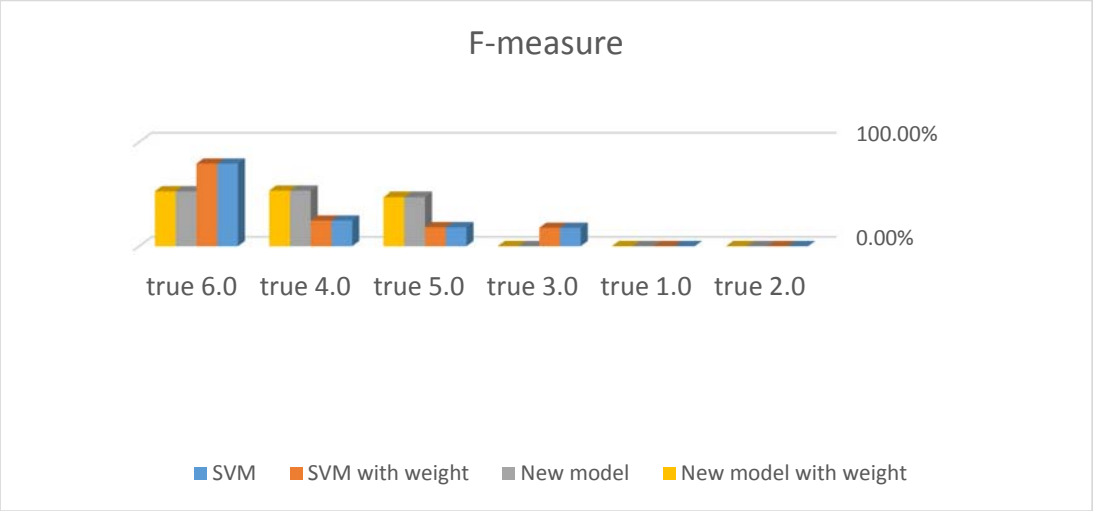


Fig. 6-9:F-measure of Dermatology Dataset

- **Balance Scale Imbalanced Multiclass Dataset**

As it mentioned previously, the model is not applicable to this kind of dataset due to the nature of the data that cannot be rebalanced utilizing the suggested Grouping Algorithm because the number of the classes that include the rare examples is only one. So, it failed to reorganize the examples in such a way that differs from its original distribution in their classes.

- **Glass Identification Imbalanced Multiclass Dataset**

Figure 6-10, Figure 6-11 and Figure 6-12 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for **Glass Identification** dataset respectively.

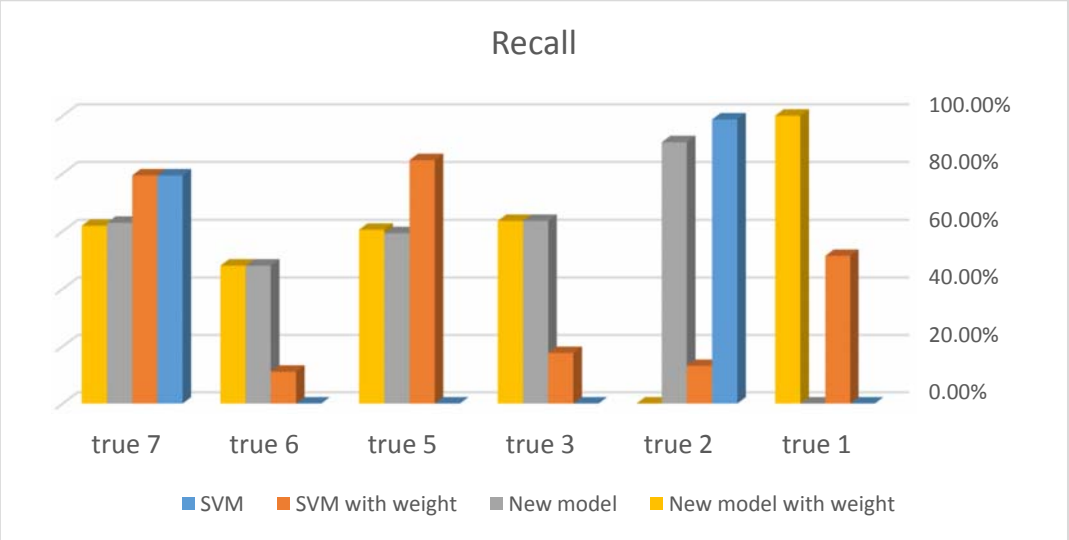


Fig. 6-10:RECALL of Glass Identification Dataset

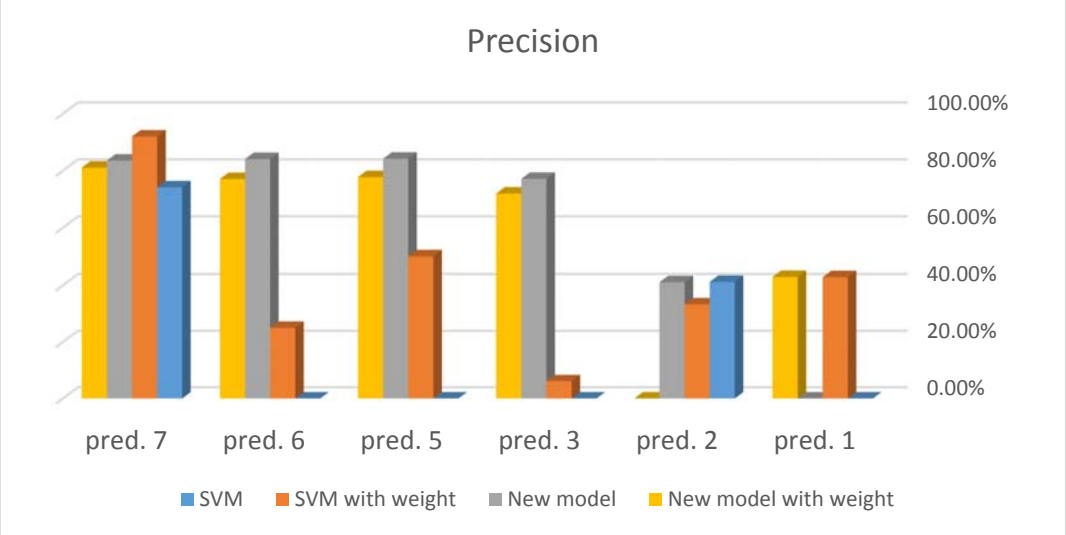


Fig. 6-11:PRECISION of Glass Identification Dataset

Fig. 6-12:F-measure of Glass Identification Dataset

- **Thyroid0387 Disease Imbalanced Multiclass Dataset**

Figure 6-13, Figure 6-14 and Figure 6-15 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for **Thyroid0387** dataset respectively.

Fig. 6-13:RECALL of Thyroid0387 Disease Dataset

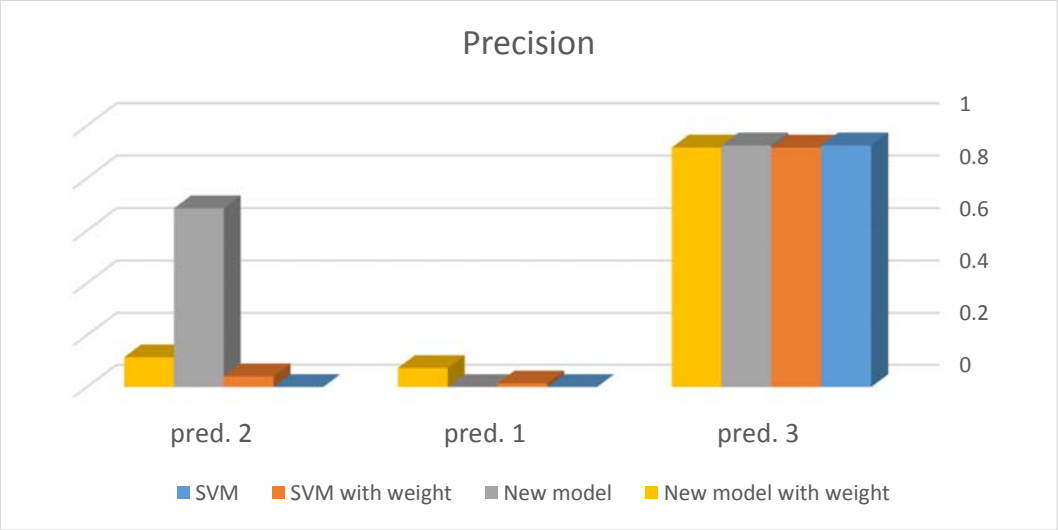


Fig. 6-14:PRECISION of Thyroid0387 Disease Dataset

Fig. 6-15:F-measure of Thyroid0387 Disease Dataset

- **Ecoli Imbalanced Multi-class Dataset**

Figure 6-16, Figure 6-17 and Figure 6-18 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for **Ecoli** dataset respectively.

Fig. 6-16:RECALL of Ecoli Disease Dataset

Fig. 6-17:Precision of Ecoli Dataset

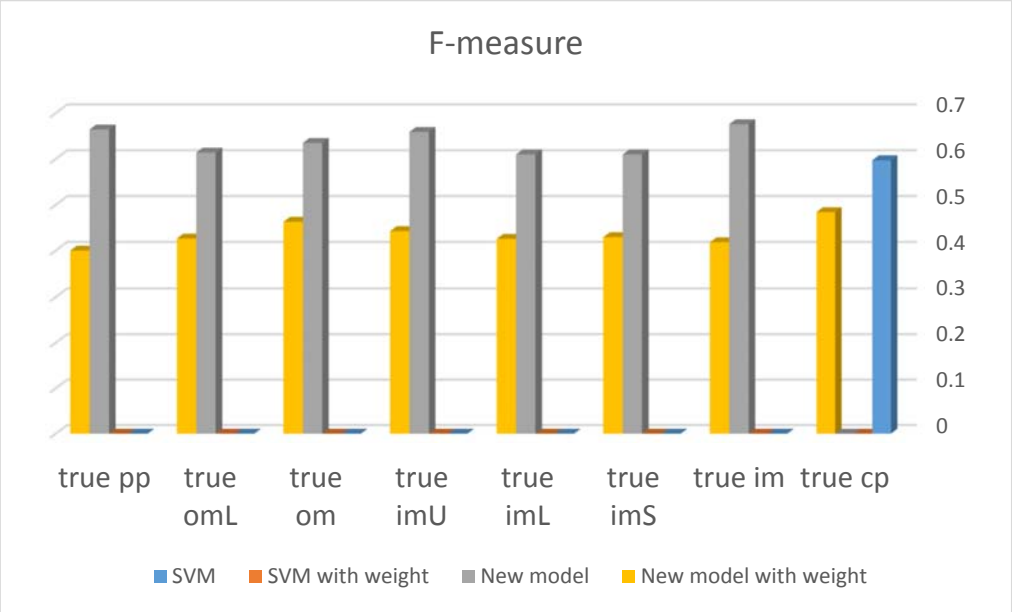


Fig. 6-18:: F-measure of Ecoli Dataset

- **Page Blocks Multi-class Imbalanced Dataset**

Figure 6-19, Figure 6-20 and Figure 6-21 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for **Page Blocks** dataset respectively

Fig. 6-19:F-measure of Page Blocks Dataset

Fig. 6-20:F-measure of Page Blocks Dataset

Fig. 6-21:F-measure of Ecoli Dataset

- **Statlog (Shuttle)**

Figure 6-22, Figure 6-23 and Figure 6-24 present the results of applying the four classification machines (Support Vector Machine without/with weight and the suggested Multi-stages model without/ with weight) considering the Recall, Precision and F-measure respectively for **Statlog (Shuttle)** dataset respectively.

Fig. 6-22:F-measure of Shuttle Dataset

Fig. 6-23:F-measure of Shuttle Dataset

Fig. 6-24:F-measure of Shuttle Dataset

Table 6-1, Table 6-2 and Table 6-3 demonstrate the results of applying the four classification methods (SVM, SVM with weight, the new model without weight and the proposed model with weight) considering (Overall Accuracy, G-mean, MFM and Kappa) respectively. The highlighted cells in the tables refer to the best results obtained.

Table 6-1:Overall Accuracy of the Four Methods

	SVM	SVM with weight	New model	New model with weight	# Class	IR
new-thyroid	0.9444	0.9448	0.90751445	0.936962751	3	4.84
dermatology	0.2074	0.2074	0.35684987	0.356849877	6	5.55
balance	0.8735	0.4671	NA	NA	3	5.88
glass	0.4578	0.3918	0.53057199	0.520231214	6	8.44
yeast	0.312	0.0101	0.36966926	0.199704724	10	23.15
thyroid	0.925	0.3833	0.84903381	0.285365854	3	36.94
ecoli	0.4257	0	0.49383949	0.311157311	8	71.5
pageblocks	0.9161	0.9197	0.76092545	0.765447667	5	164
shuttle	0.9936	0.9807	0.98568254	0.978011239	5	853

Table 6-2:G-mean of the Four Methods

	SVM	SVM with weight	New model	New model with weight
new-thyroid	0.8762	0.930466667	0.8951461	0.943216374
dermatology	0.34263	0.3426333	0.4524731	0.452473186
balance	0.63193	0.614566667	NA	NA
glass	0.29665	0.4288	0.5408611	0.556551428
yeast	0.1	0.04286	0.6266989	0.284639661
thyroid	0.33333	0.2720333	0.4688644	0.484863907
ecoli	0.125	0	0.6530398	0.429618259
pageblocks	0.29084	0.43158	0.4762204	0.532703933
shuttle	0.69054	0.75558	0.9789389	0.985885813

Table 6-3:MFM for the Four Methods

	SVM	SVM with weight	New model	New model with weight
new-thyroid	0.918893	0.926225555	0.896449491	0.935491518
dermatology	NA	NA	NA	NA
balance	NA	0.471393749	NA	NA
glass	NA	0.396901073	NA	NA
yeast	NA	NA	NA	0.114735118
thyroid	NA	0.216810724	NA	0.291046706
ecoli	NA	NA	NA	0.436091687
pageblocks	NA	NA	0.569918667	0.611539252
shuttle	NA	NA	0.987340155	0.9813908

Table 6-4:kappa for the Four Methods

	SVM	SVM with weight	New model	New model with weight
new-thyroid	0.868	0.88	0.855	0.902
dermatology	0.089	0.089	0.259	0.249

balance	0.765	0.311	NA	NA
glass	0.195	0.237	0.427	0.434
yeast	0	0	0.329	0.128
thyroid	0	0	0.373	0.022
ecoli	0	0	0.441	0.204
pageblocks	0.289	0.422	0.467	0.501
shuttle	0.982	0.947	0.98	0.97

Table 6-5: The highest Score

	Overall Accuracy	**G-mean**	**MFM**	**kappa**
SVM	3	1	0	2
with weight	2	0	2	0
New model	4	3	1	4
New model with weight	1	6	4	3

Figure 6-25, Figure 6-26 and Figure 6-27 demonstrate the results of applying the four methods (SVM, SVM with weight, the new model without weight and the proposed model with weight).

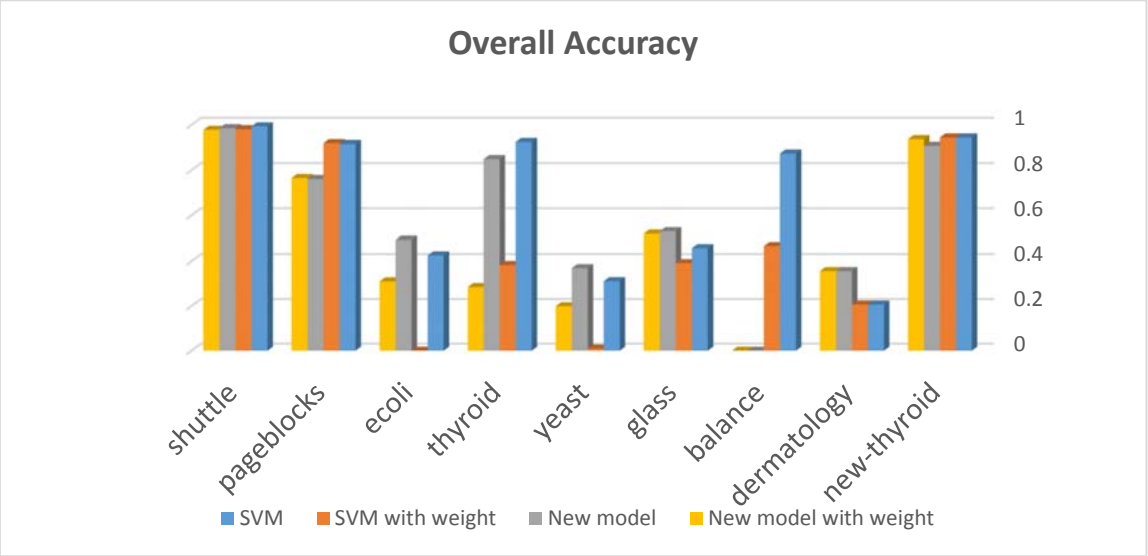


Fig. 6-25:Overall Accuracy of the four methods

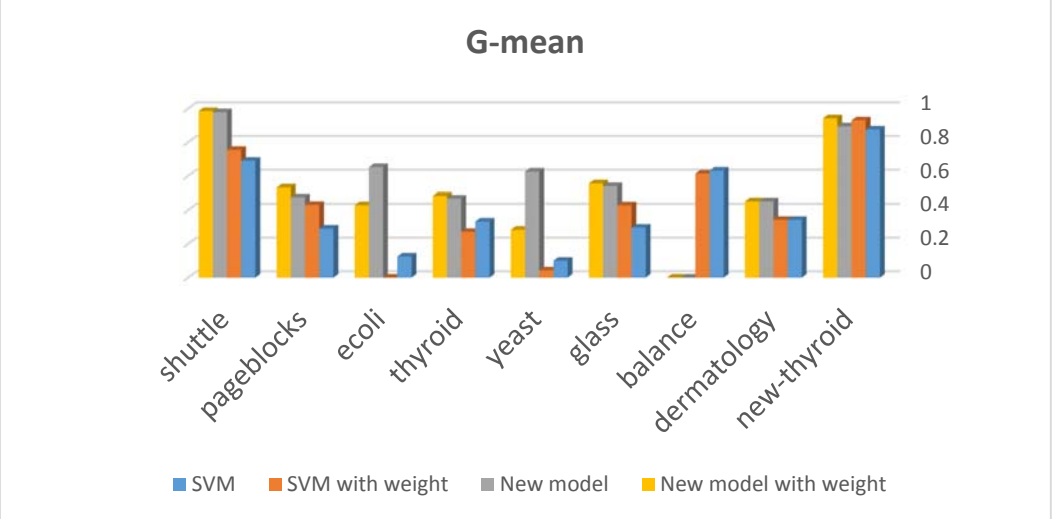


Fig. 6-26:G-mean of the four methods

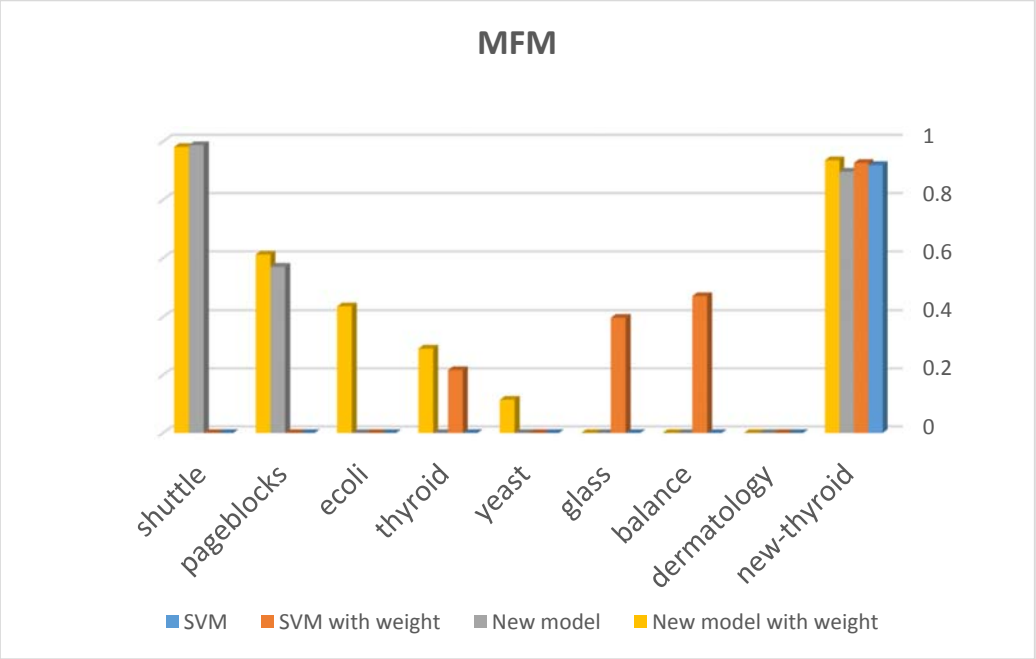


Fig. 6-27:MFM for the four methods

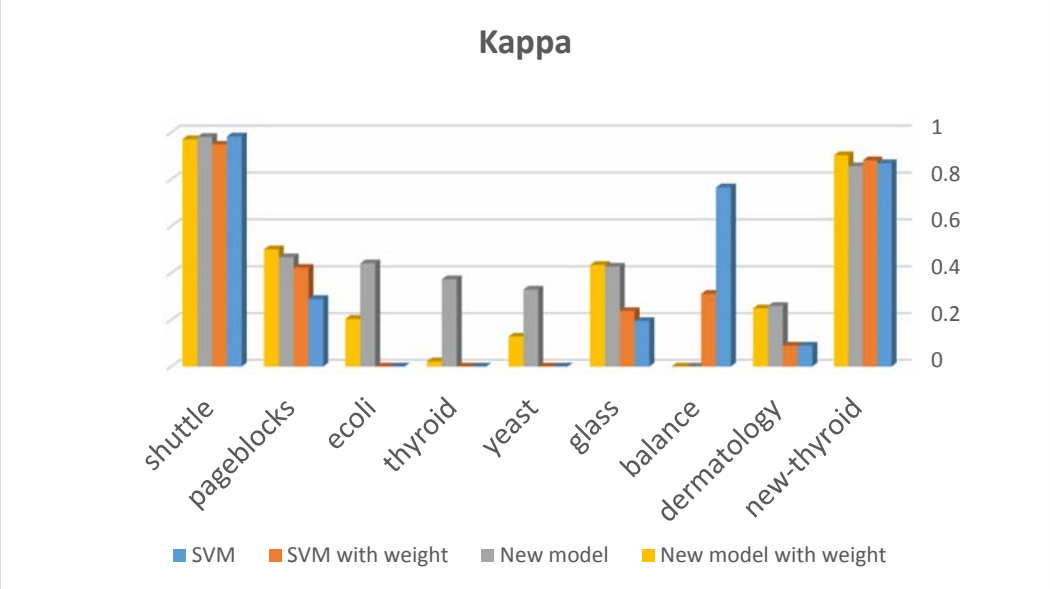


Fig. 6-28:Kappa for the four methods

Generally speaking, we notice that the results show that the performance of the proposed hierarchical method produces the best results.

When applying the proposed hierarchical model without weight, it achieves the best results in 4 out of 9 datasets in terms of Accuracy and kappa. When empowered with weight it presents the best on 6 of 9 datasets in terms of G-mean, 4 of 9 datasets considering MFM but they vary regarding the OVERALL ACCURACY.

The high performance in terms of G-mean also shows that it is good at the classification of minority class while as good as other methods for classification of majority class (can be infer from G-mean and kappa results). Average results over the 9 datasets also show that the proposed method is the best method for the four metrics.

Regarding the Overall Accuracy, we notice that the model works better as the number of the classes increases; considering the datasets Yeast, Ecoli, Glass Dermatology which have 10,8,6,6 classes respectively, the results are better when comparing with the datasets New-Thryroid, Thyroid, Pageblocks and Shuttle which have 3,3,5 classes respectively - less number of classes-.

The results also demonstrate that using the suggested hierarchical model fails in imbalance multiclass learning in a certain situation. Considering dataset 3 (Balance), it is incapable of applying the Grouping algorithm to redistribute the instances in new artificial groups. Regarding the way the algorithm works, the new groups are identical with the original classes. So, in this case the model is not applicable for such dataset. The reason for this seems to be the extreme degree of the imbalance ratio of classes' number in this dataset.

To more consolidate the model performance, different weights are added to the minority classes during learning process, but we noticed that they do not provide very high advantages for its performance. For example, the Recall of Yeast dataset when applying the model without supporting it with weights is better from its value when supplying the model with them. The same situation we get considering Ecoli dataset in terms of Recall, Precision and F-measure. We also observe that the Recall and F-measure are very similar in Glass dataset –as another instance -. Conversely, the MultiSVM is improved significantly when adding the weights during learning.

The optimal weights are in various ranges for different problems. They are decided by the proportion of the corresponding class examples within the whole data set. It can be given as:

$$\text{Weight of class}_i = \text{total sample} / (\text{number of class} * \text{sample of class}_i)$$

In regard to another perspective, the model performance dose not affected by the increasing the number of the dataset features; the results of applying the model over the New-Thyroid and Thyroid datasets -which are similar in the classes number but differ in both imbalance ratio and the number of the features - show that using the suggested hierarchical model provides advantages in both datasets in spite of the difference in their features number.

7. CHAPTER SEVEN: Conclusions

7.1. Conclusions

7.1.1. Summary of the Thesis

This research introduces the concept of learning from Imbalanced Multiclass data which is produced by many sensitive real applications. The more we pay concern to develop the classification systems of this kind of data, the more we can utilize the machines in such effective way to assist in critical fields in our life and future. It can help – for instances - in detecting rare diseases or capturing infrequent kinds of networks attacks, or discovering uncommon weld flaw in the making a nuclear weapon effectively and more.

The research problem was to find a method to deal with such data that is capable of supporting with accurate classification results, meanwhile it keeps the simplicity in its designing and implementation. So, the main aim of this research was identified in **Chapter One** which was getting more precise results of the classification process of such data.

This aim led the researcher to conduct an extensive literature review on the most widely used techniques to deal with the considering problem (presented in **Chapter Two**), thereby identifying the challenges, solutions and possible opportunities in the literature (presented at the end of Chapter Two). The vital fact from this survey is that, till now there is no best technique for problems for all situations and datasets. By the end of this survey, one of the objectives of this research was obtained. This objective was reviewing the different solutions of treating Imbalanced Multiclass data and address their advantages and shortcomings.

Chapter Three satisfied the next objective which was developing a multi-stages model for the classification process using Binary and Multi-Class Support Vector Machines. It was chosen due to its solid mathematical background regarding Imbalanced Multi-Class data.

For the experiments setup, nine popular imbalanced multi-class datasets were selected from U.C.I., they were from different fields such as physics, biology, and medicine. Their format, characteristics and imbalance limits were investigated in **Chapter Four** to be able to achieve the objective of applying the suggested model. So, in the same chapter, each dataset has been examined by four machines: SVM with and without weight and the proposed model with and without weight so as to compare the proposed model with the two of strong state-of-art solutions.

In order to be able to investigate the overall performance a small review of the performance metrics dedicated for evaluating classifiers that learn from binary imbalanced, Multiclass and Multiclass imbalanced data was introduced in **Chapter Five** to stand on a solid base of knowledge to choose the most suitable metrics to be utilized so as to satisfy the objective of investigating the model performance.

Chapter Six demonstrates the classification results and introduces a discussion about it involving some explanation, justifications and comparisons for different utilized methods and their performance regarding some selected metrics.

7.1.2. Findings of the Thesis

- The experiments show that the new hierarchical model enhances the classification results comparing with the classification results of

mentioned state-of art solution, even when empowered with weight for minority instances, considering four different performance metrics.

- The model Grouping Algorithm is successful in classifying imbalanced data sets. It performs well even when the ratio between minority and majority samples is high, but it failed when the minority classes number is little comparing with the majority classes number.

7.1.2.1. **The Model Advantages are:**

- It does not require any data pre-processing step as many other solutions need.
- Handle the imbalance nature of the data simply without computational efforts or algorithmic adjustment. It even does not need to be empowered with any cost function as results shows.
- It does not use any fixed hierarchy based on features and/or classes. Unlike the common hierarchical methods which use supervised learning, the suggested hierarchical model is grounded on a Grouping Algorithm that redistributes the instances artificially basing on the least difference between the new created groups in their sizes.
- For each multiclass SVM classifier at each stage, the number of classes is less than the overall number of the dataset classes, so the classifier offers satisfactory results than dealing with the dataset as all.
- It performs well when dealing with large numbers of classes.
- Although it groups various heterogeneous kinds of classes in one group, it exploits the black box nature of the Support Vector Machine which could be considered a benefit in our case.

7.1.2.2. **The Model Disadvantages are:**

- It also deals poorly with the dataset that owns little number of classes that could not be decomposed into groups of nearly balanced numbers of examples.
- Naturally, it owns the flaws of hierarchical classification models that cannot produce their final classification result unless the path from the root to the final leaf is passed, which may consume more time.

7.2. Future Suggested Works:

- The results obtained in this research can be tested by using other distinct types of Multiclass Support Vector Machine or other data mining tools of classifications such as Neural Networks or ensemble techniques.
- In order to better evaluate the proposed model in this research, it has to be implemented over real life data.
- It also can be tested for large scale of data.
- As well as test data with a very large number of classes.
- A new strategy of grouping the classes instead of basing on the least difference between the created groups could be tried and tested.

Bibliography

[1] H. He and Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications, vol. first edition, H. H. a. Y. Ma, Ed., John Wiley & Sons, Inc, 2013.

[2] H. He and E. Garcia, "Learning from Imbalanced Data," *IEEE Transactions On Knowledge And Data Engineering,,* vol. 21, no. 9, September 2009.

[3] J. Taeho and J. Nathalie, "Class imbalances versus small disjuncts," vol. 6, p. 40–49, 2004.

[4] M. Kubat and M. Stan, "Addressing the curse of imbalanced training sets: One-sided selection," vol. 97, p. 179–186, 1997.

[5] Hart, E. Pete, J. Nils, Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," vol. 4, p. 100–107, 1968.

[6] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," vol. 2101, p. 63–66, 2001.

[7] N. Chawla, K. Bowyer, L. Hall and Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," vol. 16, p. 321–357, 2002.

[8] H. H, B. Y, G. E A and L. S, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," (Hong Kong, China, 2008.

[9] N. Japkowicz, "Learning from imbalanced data sets: A comparison of various strategies," vol. 68, 2000.

[10] B. GE, P. RC and M. MC, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter,* vol. 6, p. 20–29, 2004.

[11] Wang, Shuo, T. Ke and Y. Xin, "Diversity exploration and negative correlation learning on imbalanced data sets," in *International Joint Conference on IEEE*, Atlanta, GA, 2009.

[12] C. NV, L. A, L. Hall and B. KW, "Smoteboost: Improving prediction of the minority class in boosting," *knowledge Discovery in Database PKDD,* pp. 107-119,

2003.

[13] S. C, K. T, H. JV and N. A, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* vol. 40, p. 185–197, 2010.

[14] H. G. a. H. L. r, "Learning from imbalanced data sets with boosting and data generation: The Databoost-IM approach," *SIGKDD Explorations Newsletter,* vol. 6, p. 30–39, 2004.

[15] J. W. a. Z.-H. Z. X.-Y. Liu, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics,* vol. 39, p. 539–550, 2009.

[16] J. L. a. N. A. B. Zadrozny, "Cost-sensitive learning by cost-proportionate example weighting," 2003.

[17] P. Domingos, "MetaCost: A general method for making classifiers cost-sensitive," 1999.

[18] K. M. Ting, "A comparative study of cost-sensitive boosting algorithms," Stanford, CA, 2000.

[19] P. V. a. M. Jones, "Fast and robust classification using asymmetric AdaBoost and a detector cascade," Cambridge.

[20] S. C. B. M. D. M. a. J. M. R. J. Wu, "Fast asymmetric learning for cascade face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, p. 369–382, 2008.

[21] S. Y, W. AKC and W. Y, "Parameter inference of cost-sensitive boosting algorithms," Leipzig, Germany, 2005.

[22] S. J. S. J. Z. a. P. K. C. W. Fan, "AdaCost: Misclassification costsensitive boosting," Bled, Slovenia, 1999.

[23] J. N, "Class imbalances versus small disjuncts," 2004.

[24] C. Elkan, "The foundations of cost-sensitive learning,," 2001.

[25] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," 2003.

[26] M. Z. K. a. I. Kononenko, "Cost-sensitive learning with neural networks," 1998.

[27] Zhou, L. XY and H. Z, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, p. 63–77, 2006.

[28] G. R. I. W. a. M. J. Pazzani, "Adjusted probability naive Bayesian induction," Brisbane, Australia, 1998.

[29] G. F. a. F. Roli, "Support vector machines with embedded reject option," Niagara Falls, Canada, 2002.

[30] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," Cambridge,MA,MIT Press, 1999.

[31] J. T. Kwok, "Moderating the outputs of support vector machine classifiers," *IEEE Transactions on Neural Networks,* vol. 10, p. 1018–1031, 1999.

[32] V. VN, The Nature of Statistical Learning Theory, Springer, 1995.

[33] N. J. a. S. Stephen, "The Class Imbalance Problem: A Systematic Study," *Intelligent Data Analysis,* vol. 6, pp. 429- 449,, 2002.

[34] B. R. a. A. Kowalczyk, "Extreme Re-Balancing for SVMs: A Case Study," *ACM SIGKDD Explorations Newsletter,* vol. 6, pp. 60-69, 2004.

[35] S. K. a. N. J. R. Akbani, *Applying Support Vector Machines to Imbalanced Data Sets,* vol. 3201, 2004, pp. 39-50.

[36] Z. C. L. Z. a. B. H. Qiong Gu, " Data mining on imbalanced data sets," 2008.

[37] J. H. a. L. G. S. Ertekin, "Active learning for class imbalance problem," (Amsterdam, The Netherlands, 2007.

[38] J. H. L. B. a. L. G. S. Ertekin, "Learning on the border: Active learning in imbalanced data classification," Lisbon, Portugal, 2007.

[39] Y. F. a. R. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Computer and System Sciences,* vol. 55, pp. 119-139, 1997.

[40] A. K. a. B. Raskutti, "One class SVM for yeast regulation prediction," *SIGKDD*

Exploration Newsletters, vol. 4, p. 99–100, 2002.

[41] E. T. a. J. N., "A Recognition- based Alternative to Discrimination-based Perceptrons," London, UK, 2000.

[42] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[43] X.-Y. L. Zhi-Hua Zhou, "On Multi-Class Cost-Sensitive Learning," 2006.

[44] R. F. &. C. BEYAN, "Classifying Imbalanced Data Sets Using Similarity Based Hierarchical Decomposition," *Journal Pattern Recognition © ACM,* vol. 48, pp. 1653-1672, May 2015.

[45] I. P. E. G.-R. A. Rohit Babbar, "On Flat versus Hierarchical Classification in Large-Scale Taxonomies," vol. 26, 2013.

[46] Mohamed Aly, "Survey on Multiclass Classification Methods," Caltech, USA, 2005.

[47] M. A. F. E. B. H. B. a. F. H. Galar, "Aggregation Schemes for Binarization Techniques. Methods' Description," EUROFUSE'09: Workshop on Preference Modelling and Decision Analysis, 2009.

[48] L. X. a. A. M. Stephen Boyd, "Notes on Decomposition Methods, Notes for EE392o," Stanford University, Stanford,USA, October 1, 2003.

[49] S. V. G. W. Amal S. Ghanem, " Multi-Class Pattern Classification in Imbalanced Data," 2010.

[50] S. Wang, *Ensemble Divesity For Class Imbalance,* Birmingham: University of Birmingham, July 2011.

[51] R. R. a. A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Researches,* vol. 5, p. 101–141, Dec. 2004.

[52] T. H. a. R. Tibshiran, "Classification by pairwise coupling," *Annals of Statistics,* vol. 26, p. 451–471, Apr,1998.

[53] A. F. E. B. B. H. Mikel Galar, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," vol. 44, no. 8, p. 1761–1776, August 2011.

[54] D. G. a. Y. D. A. C. Tan, "Multi-class protein fold classification using a new ensemble machine learning approach," *Genome Inf.,* vol. 14, p. 206–217, 2003.

[55] T. G. D. a. G. Bakiri, "Solving multiclass learning problems via error correcting output codes," *Journal of Artificial Intelligence Research ,* vol. 2, pp. 263-286, 1995.

[56] C. G. L. Z. N. Cesa-Bianchi, "Incremental algorithms for hierarchical classification," *The Journal of Machine Learning Research 7,* vol. 7, pp. 31-54, 2006.

[57] N. C. J. S.-T. John Platt, "Large Margin DAG's for Multiclass Classification," January 1999.

[58] J. G. M. M. C. S. Kumar, "Hierarchical fusion of multiple classifiers for hyperspectral data analysis," *Pattern Analysis & Applications,* vol. 5, pp. 210-220, 2002.

[59] Y. K. T. H. Chiang, "Hierarchically SVM classification based on support vector clustering method and its application to document categorization," *Expert Systems with Applications,* vol. 33, pp. 627-635, 2007.

[60] B. L. L. a. J. K. K. Chen, "Efficient classification of multi-label and imbalanced data using min-max modular classifiers," Vancouver, BC, 2006.

[61] M. J. d. J. a. F. H. Alberto Fernandez, "Multi-class imbalanced data-sets with linguistic fuzzy rule based classification systems based on pairwise learning.," in *Computational Intelligence for Knowledge-Based Systems Design*, vol. 6178, R. K. a. F. H. Eyke Hüllermeier, Ed., Dortmund, Germany, Springer Berlin Heidelberg, June 28 - July 2010, pp. 89-98.

[62] X. Y. Shuo Wang, "Multi-Class Imbalance Problems: Analysis and," vol. 42, no. 4, pp. 1119 - 1130, 2012.

[63] X. L. L. C. a. K. A. Xing-Ming Zhao, "Protein classification with imbalanced data," *Proteins: Structure, Function, and Bioinformatics,* vol. 70, pp. 1125-1132, March 2008.

[64] D. G. a. Y. D. Aik Choon Tan, "Multi-class protein fold classification using a new ensemble machine learning approach," *Genome Informatics,* vol. 14, pp. 206-217, 2003.

[65] S. V. G. W. Amal S. Ghanem, "Multi-Class Pattern Classification in Imbalanced Data," Washington, DC, USA, 2010.

[66] T. W. Liao., "Classification of weld flaws with imbalanced class data," *Expert Systems with Applications,* vol. 35, pp. 1041-1052, October, 2008.

[67] A. L. a. H. S., "To Combat Multi-class Imbalanced Problems by Means of Over-sampling and Boosting Techniques," *Soft Computing,* vol. 19, no. 10.1007/s00500-014-1291-z, pp. 3369-3385, Dec 2015.

[68] P. a. W. K. Jeatrakul, "Enhancing classification performance of multi-class imbalanced data using the OAA-DB algorithm," Brisbane, Australia, 10 - 15 June 2012.

[69] J. M. S. R. M. V. a. G. A. C. Roberto Alejo, "The multi-class imbalance problem: Cost functions with modular and non-modular neural networks," in *The Sixth International Symposium on Neural Networks (ISNN 2009)*, vol. 56, Y. S. T. H. a. Z. Z. Hongwei Wang, Ed., Wuhan, China, Springer Berlin Heidelberg, 2009, pp. 421-431.

[70] B. Z. a. J. L. N. Abe, "An iterative method for multi-class cost sensitive learning," Seattle, WA, USA, 2004.

[71] R. P. D. Thomas C.W. Landgrebe, "Approximating the multiclass ROC by pairwise analysis," *Pattern Recognition Letters,* vol. 28, p. 1747–1758, 1 October 2007.

[72] N. N. I. T. o. Chih-Wei Hsu and Chih-Jen Lin, "A Comparison of Methods for Multi-class Support Vectore Machines," *IEEE Transactions on Neural Networks,* vol. 13, March 2002.

[73] H. W. G. O. a. L. F. Y. Murphey, "OAHO: an effective algorithm for multi-class learning from imbalanced data," Orlando, FL, USA, Aug. 2007.

[74] M. O. T. Li, "Music genre classification with taxonomy," Philadelphia, PA, 2005.

[75] J. Z. V. H. F. Wu, "Learning classifiers using hierarchically structured class taxonomies," in *Abstraction, Reformulation and Approximation*, vol. 3607, J. Z. a. L. Saitta, Ed., Airth Castle , Scotland: Proceedings of the Symposium on Abstraction, Reformulation, and Approximation, Springer Berlin Heidelberg, 2005, pp. 313-320.

[76] S. a. M. A.Ramanan, "Unbalanced Decision Trees for Multi-class Classification," Penadeniya, Aug. 2007.

[77] M. M. C. J. G. Y. Chen, "Integrating support vector machines in a hierarchical output space decomposition framework," 2004.

[78] Q. Q. N. V. C. a. Z.-H. Z. Ryan Hoens, "Building Decision Trees for the Multi-class Imbalance Problem," in *Advances in Knowledge Discovery and Data Mining*, vol. 7301, S. C. K. H. B. Pang-Ning Tan, Ed., Kuala Lumpur, Springer Berlin Heidelberg, May 29-June 1, 2012, pp. 122-134.

[79] ZHANG, B. LUO and Yun, "Hierarchical Classification for Imbalanced Multiple Classes in Machine Vision Inspection," 2007.

[80] S. U. B. Epshtein, "Feature hierarchies for object classification," 2005.

[81] B. J. B. B. F. F. P. X. Huang, "Underwater Live Fish Recognition using a Balanced-Guaranteed Optimized Tree," in *Computer Vision – ACCV 2012*, vol. 7724, Y. M. ,. J. M. R. H. Kyoung Mu Lee, Ed., Daejeon , Proceedings of 11th Asian Conference on Computer Vision (ACCV),Springer Berlin Heidelberg, 2013, pp. 422-433.

[82] D. K. O. B. C. Freeman, "Joint feature selection and hierarchical classifier design," Anchorage, AK, 2011.

[83] E. L. a. G. P. J. Kindermann, " Multi-class classification with error correcting codes," German National Research Center for Information Technology (GMD), Sankt Augustin, 2000.

[84] C. C. a. V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, p. 273–297, 1995.

[85] I. G. a. V. V. B. Boser, "A training algorithm for optimal margin classifiers," Pittsburgh, PA, USA, 1992.

[86] N. C. a. J. S.-T. L. I. A. J. C. Platt, "Large margin DAGs for multiclass classification," 2000.

[87] J. W. a. C. Watkins., "Multi-class support vector machines," Brussels, 1999.

[88] E. M. a. E. Alpaydin, "Support vector machines for multi-class classification," 1999.

[89] Y. Guermeur, "Combining Discriminant Models with new Multi-Class SVMs," HAL, 2000.

[90] K. C. a. Y. Singer, "Ultraconservative online algorithms for multiclass problems.," School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel, 2001.

[91] C.-C. C. a. C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," vol. 2, no. 3, April 2011 .

[92] H. Surks MI, "Age-sepecific distribution of serum thyrotropin and antithyroid antibodies in the US population:implications for the prevalence of subclinical hypothyroidism," *J. Clin Endocrinol Metab,* vol. 92, pp. 4575-82, December 2007.

[93] Random House, *Random House Webster's Unabridged Dictionary,* 2001, p. 573.

[94] J. H.-O. a. R. M. C. Ferri, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters,* vol. 30, p. 27–38, 2009.

[95] R. C. a. A. Niculescu-Mizil, "Data mining in metric space: An empirical analysis of supervised learning performance criteria,," Seattle, WA, 2004.

[96] N. J. a. M. Shah, Evaluating Learning Algorithms: A Classification Perspective, New York: Cambridge University Press, June 2014.

[97] V. K. a. R. A. M.V. Joshi, "Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements," San Jose, CA, 2001.

[98] F. P. a. T. Fawcett, " Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions," 1997.

[99] W. G.M, "Mining with rarity: a unifying framework," *SIGKDD Explorations ,* vol. 6, 2004.

[100] R. C. H. a. S. M. M. Kubat, "Machine learning for the detection of oil spills in satellite radar images," *Machine Learning,* vol. 30, p. 195–215, 1998.

[101] A. C. L. A. C. P. L. F. C. A. A. F. Eduardo P. Costa, "A Review of Performance Evaluation Measures for Hierarchical Classifiers," 2007.

[102] M. Lawrence, *A balanced approach to the multi-class imbalance problem,* Iowa: Iowa State University,Industrial and Manufacturing Systems Engineering

Department, http://lib.dr.iastate.edu/etd/13537, 2013, p. 13537.

[103] S. Y, K. MS and W. Y, "Boosting for learning multiple classes with imbalanced class distribution," Washington, DC, USA, 2006.

[104] X. L. a. Z. Zhou, "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," vol. 18, no. 1, pp. 63-77, Jan. 2006.

[105] D. G. a. Y. D. Aik Choon Tan, "Multi-class protein fold classification using a new ensemble machine learning approach," vol. 14, pp. 206-217, 2003.

[106] J. Richard Landis and Gary G. Koch, "The Measurement of Observer Agreement for Categorical Data," vol. 33, no. 1, pp. 159-174, Mar 1977.

[107] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," Technical Report HPL,HP Labs, 2003.

[108] Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters,* vol. 27, pp. 861-874, 2006.

[109] J. E. F. Richard M. Everson, "Multiclass ROC analysis from a multi-objective optimization perspective," vol. 27, no. 8, pp. 918-92, June 200.

[110] D. H. a. R. Till, "A Simple Generalization of the Area under the ROC Curve to Multiple Class Classification Problems," *Machine Learning,* vol. 45, pp. 171-186, 2001.

[111] F. P. a. P. Domingos, *Well-Trained Pets: Improving Probability Estimation Trees,,* New York: Stern School of Business, New York University, 2000.

[112] a. R. P. D. Thomas Landgrebe, "A simplified extension of the Area under the ROC to the multiclass domain"," Novmber, 2006.

[113] A. G. B. S. J.-C. M. Sidney D´Mello, "Affective Computing and Intelligent Interaction," TN,USA, October 2011.

[114] R. J. K. A. B. Hyndman, "Another Look at Measures of Forecast Accuacy," vol. 22, pp. 679-688, 2006.

[115] K. H. B. J. A. C. Henry Carrillo1, "Probabilistic performance evaluation for multiclass classification using the posterior balanced accuracy," 2013.

[116] P. G. L. a. S. M. Ipeirotis, "Probe, count, and classify: categorizing hidden web databases," NY, USA, 2001.

[117] A. C. L. A. C. P. L. F. C. A. A. F. Eduardo P. Costa, "Evaluation Measures for Hierarchical Classifiers," 2007.