

EVOLUTIONARY DEEP BELIEF NETWORK WITH  
BOOTSTRAP SAMPLING FOR IMBALANCED CLASS  
DATA

BY

A'INUR A'FIFAH AMRI

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

2019

EVOLUTIONARY DEEP BELIEF NETWORK WITH  
BOOTSTRAP SAMPLING FOR IMBALANCED CLASS  
DATA

BY

A'INUR A'FIFAH AMRI

A thesis submitted in fulfilment of the requirement for the  
degree of Master  
Computer Science

Kulliyyah of Information and Computer Technology  
International Islamic University Malaysia

JANUARY 2019

## ABSTRACT

Imbalanced class data is a frequent problem faced in classification task. Imbalanced class occurs when the classes in the dataset has a huge distribution gap between them. The class with the most instances is called the majority class, while the class with the least instances is called the minority class. it caused the result to be skewed towards the majority class instead. Common techniques to overcome or minimize the negative effect of imbalanced class data are data sampling, algorithm modification or hybrid. Deep learning algorithm is a state-of-the-art part of the machine learning algorithms. It is popularized due to better performances when handling complex and high dimensional data. Deep belief network (DBN) is an example of a deep learning algorithm. It is an intricate form of artificial neural network (ANN). It has a deeper layer of neurons that prevents the network from getting stuck when learning from the inputs. It pre-trains the network using Restricted Boltzmann Machine (RBM) and implement backpropagation neural network (BPNN) as a fine-tune step. However, the training time for DBN is longer because of the layers. Also, there is very little apprehension for the exact amount of data or ideal hyperparameters setting to optimize the performance. Due to its complex and deep layers architecture, deep learning needs a lot of training data in order to give good predictions. In this thesis, an optimized DBN is proposed to control the negative outcomes caused by imbalanced class data towards the performance of the algorithm using an evolutionary algorithm. An evolutionary algorithm (EA) is incorporated to provide the optimum dropout number, learning rate, batch size and iteration number of BPNN for fine-tuning in DBN. Bootstrap sampling is also incorporated in the algorithm structure to minimize the bias of data training samples. These modifications improved the ability to predict more accurate outcomes. To evaluate the performance of Evolutionary DBN with bootstrap sampling, an experimental setup involving imbalanced class datasets are conducted. The results of Evolutionary DBN with bootstrap sampling performance is collected and documented in the form of performance metrics. The results are then compared to other machine learning algorithms such as DBN, deep neural network (DNN), BPNN and support vector machine (SVM). According to the outcomes, Evolutionary DBN with bootstrap sampling performed better than DBN and other machine learning algorithms in managing the effects of imbalanced class datasets such as accurate predictions and less partiality. The analysis of statistical tests conducted at the end of this thesis supports the conclusion of the experiment.

## الملخص

تعتبر بيانات الفئة غير المتوازنة مشكلة متكررة تواجه عملية تصنيف البيانات. ويحدث عدم التوازن عندما تكون هناك فجوة كبيرة في توزيع البيانات. يُطلق على الفئة التي تحتوي على حالات كثيرة من عدم التوازن فئة الأغلبية ، بينما تسمى الفئة التي تحتوي على حالات اقل فئة الأقلية. وتسبب في انحراف النتيجة لصالح الفئة الاغلبية. هناك تقنيات شائعة للتغلب او الحد من التأثير السلبي لبيانات الفئة غير المتوازنة وهي : اختيار العينات او تعديل الخوارزمية او طريقة الهجين. تعد خوارزمية التعلم العميق قسم من خوارزميات تعلم الاله حيث انها شائعة بسبب الاداء الجيد عند التعامل مع بيانات معقدة وعالية الابعاد. وتعتبر شبكة الوثوق العميق (DBN) أحد امثلة خوارزمية التعلم العميق وهي شكل معقد من شبكة الأعصاب الصناعية(ANN). اذ تحتوي على طبقة اعماق من الخلايا العصبية والتي تحافظ على الاداء الجيد عند التعلم من المدخلات. وتقوم بتدريب مسبق للشبكة باستخدام آلة بولتزمان المقيدة (RBM) والشبكة العصبية العكسية (BPNN) كخطوة ضبط دقيقة. ومع ذلك فإن شبكة الوثوق العميق (DBN) تحتاج الى وقت طويل في عملية التمرين بسبب وجود الخلايا. ايضا تحتاج الى عملية ضبط واعداد مثالي لعناصرها حتى تحصل على النتيجة المطلوبة. كما ان التعلم العميق تحتاج الى بيانات كثيرة من اجل عملية تدريب البيانات لتحصل على توقعات جيدة وذلك بسبب التعقد والعمق في طبقاتها. هذه الأطروحة تقترح تحسين شبكة الوثوق العميق (DBN) للتحكم في النتائج السلبية الناجمة عن بيانات الفئة غير المتوازنة من خلال الخوارزمية التطويرية (EA). تقوم هذه الخوارزمية بايجاد القيمة المثلى لكل من : التسرب , معدل التدريب , حجم الدفعة وعدد مرات التكرار. كما انه تم دمج عملية عينات التمهيد في بنية الخوارزمية لتقليل تحيز عينات تدريب البيانات. تهدف هذه التعديلات الى تحسين عملية التنبؤ بالنتائج. ومن اجل تقييم التعديلات المقترحة للخوارزمية اجريت تجارب تحتوي على بيانات فئة غير متوازنة. وتم جمع النتائج وتوثيقها على هيئة مقياس اداء. ايضا تم مقارنة نتائج الخوارزمية مع خوارزميات اخرى مثل الشبكة العصبية العميقة (DNN) ونشر شبكة العصبية (BPNN) ودعم شاحنات النقل (SVM) وشبكة الوثوق العميق (DBN). وفقا للنتائج فان شبكة الوثوق العميق (DBN) باندماج مع عينات التمهيد حقق اداء أفضل من شبكة الوثوق العميق (DBN) وبقية خوارزميات التعلم الالي في التحكم بالتأثيرات الناتجة عن بيانات الفئة غير المتوازنة مثل دقة التنبؤ واقل تحيز. وقد تم اجراء تحليل الاختبارات الإحصائية في نهاية الاطروحة حتى يدعم التجارب.

## APPROVAL PAGE

I certify that I have supervised and read this study and that in my opinion; it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master Computer Science.

.....  
Amelia Ritahani Ismail  
Supervisor

I certify that I have read this study and that in my opinion; it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master Computer Science.

.....  
Raini Hassan  
Internal Examiner

.....  
Mohd Zakree Ahmad Nazri  
External Examiner

This thesis was submitted to the Department of Computer Science and is accepted as a fulfilment of the requirement for the degree of Master Computer Science.

.....  
Raini Hassan  
Head, Department of  
Computer Science

This thesis was submitted to the Kuliyyah of Information and Computer Technology and is accepted as a fulfilment of the requirement for the degree of Master Computer Science.

.....  
Abdul Wahab Abdul Rahman  
Dean, Kulliyah of Information  
and Computer Technology

## **DECLARATION**

I hereby declare that this thesis is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

A'inur A'fifah Amri

Signature:.....

Date:.....

**INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA**

**DECLARATION OF COPYRIGHT AND AFFIRMATION OF  
FAIR USE OF UNPUBLISHED RESEARCH**

**EVOLUTIONARY DEEP BELIEF NETWORK WITH  
BOOTSTRAP SAMPLING FOR IMBALANCED CLASS DATA**

I declare that the copyright holder of this thesis is A'inur A'fifah Amri.

Copyright © 2016 A'inur A'fifah Amri. All rights reserved.

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may only be used by others in their writing with due acknowledgement.
2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purpose.
3. The IIUM library will have the right to make, store in a retrieval system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by A'inur A'fifah Amri.

.....

Signature

.....

Date

## **ACKNOWLEDGEMENTS**

I would like to dedicate this piece of work to my parents, sisters, relatives and friends for their emotional support and for making me believe in myself to keep going and complete this thesis. Thank you for reminding me the importance of working hard and working sincerely.

I would like to thank my supervisor, Assoc. Prof. Dr. Amelia Ritahani Ismail for her constant support and guidance. She taught me how to navigate my journey in the academic world which in turn chart my own personal growth. From the exterior, it might seem I've only learned how to read and write journal papers and eventually this thesis, but I also learned to narrow down my focus for my thesis and discipline myself to finally get the thesis done. Due to her guidance, I've also learned to improve my way of thinking and overcome the challenges in my experiment like a researcher as opposed to merely think like a student. Not only that, I appreciate for her emotional support throughout the process of working on this thesis.



# TABLE OF CONTENTS

Abstract .....	ii
Abstract in Arabic .....	iii
Approval Page .....	iv
Declaration .....	v
Declaration Page .....	v
Copyright .....	vi
Acknowledgements.....	vii
List of Tables.....	x
List of Figures .....	xi
 <b>CHAPTER 1: INTRODUCTION .....</b>	 <b>1</b>
1.1 Study Background .....	1
1.2 Problem Statement.....	3
1.3 Research Hypotheses.....	4
1.4 Research Objectives .....	4
1.5 Research Questions .....	4
1.6 Significance and Contribution.....	5
1.7 Thesis Structure .....	5
 <b>CHAPTER 2: LITERATURE REVIEW .....</b>	 <b>6</b>
2.1 Introduction.....	6
2.2 Imbalanced Class Dataset .....	6
2.3 Data Sampling Method .....	7
2.3.1 Oversampling.....	8
2.3.2 Undersampling .....	8
2.3.3 Bootstrap Sampling .....	9
2.4 Machine Learning Algorithm .....	10
2.4.1 Backpropagation Neural Network Algorithm .....	11
2.4.2 Support Vector Machine Algorithm.....	13
2.5 Deep Learning Algorithm.....	14
2.5.1 Deep Belief Network Algorithm.....	15
2.5.2 Deep Neural Network Algorithm.....	20
2.5.3 Convolutional Neural Network Algorithm .....	22
2.5.4 Findings in Deep Learning Literature Study.....	24
2.6 Evolutionary Algorithms .....	24
2.6.1 Genetic Algorithm.....	25
 <b>CHAPTER 3: RESEARCH METHODOLOGY .....</b>	 <b>28</b>
3.1 Introduction.....	28
3.1.1 Studying the Literature.....	29
3.1.2 Selecting the Datasets for the Experiment.....	30
3.1.3 Implementing Deep Belief Network Model for the Imbalanced Class Datasets .....	31
3.1.4 Implementing Evolutionary Algorithm and Bootstrap Sampling for the Deep Belief Network .....	32
3.1.5 Analyzing the Performances of Algorithms on the Datasets	32

3.1.6	Comparing the Proposed Algorithm with Other Algorithms	33
3.1.7	Documenting the Results of the Simulation, Evaluation, Comparisons and Analysis .....	33
<b>CHAPTER 4: EXPERIMENTAL SETUP .....</b>		<b>34</b>
4.1	Introduction.....	34
4.2	Imbalanced Class Dataset .....	34
4.3	Metrics Evaluation .....	39
4.3.1	Performance Metrics.....	40
4.3.1.1	Confusion Matrix .....	41
4.3.1.2	Accuracy Rate .....	42
4.3.1.3	Weighted Mean Recall.....	43
4.3.1.4	Weighted Mean Precision.....	43
4.3.1.5	F1-Score .....	44
4.3.1.6	Area Under Curve .....	44
4.3.2	Statistical Tests .....	45
4.3.2.1	Wilcoxon Signed Rank Test.....	45
4.3.2.2	Vargha-Delaney A Test.....	46
<b>CHAPTER 5: EVOLUTIONARY DEEP BELIEF NETWORK WITH BOOTSTRAP SAMPLING .....</b>		<b>48</b>
5.1	Introduction.....	48
5.2	Deep Belief Network .....	48
5.3	Evolutionary Deep Belief Network with Bootstrap Sampling .....	54
<b>CHAPTER 6: RESULTS AND DISCUSSION.....</b>		<b>62</b>
6.1	Introduction.....	62
6.2	Performance Metrics Results .....	62
6.2.1	Confusion Matrix .....	63
6.2.2	Accuracy Rate .....	65
6.2.3	Weighted Mean Recall .....	69
6.2.4	Weighted Mean Precision .....	73
6.2.5	F1-Score .....	76
6.2.6	Area Under Curve .....	78
6.2.7	Analysis of Performance Metrics.....	82
6.3	Statistical Test Analysis .....	85
6.3.1	Wilcoxon Signed Rank Test.....	86
6.3.2	Vargha-Delaney A Test.....	89
<b>CHAPTER 7: CONCLUSION AND FUTURE WORKS .....</b>		<b>92</b>
7.1	Introduction.....	92
7.2	Conclusion .....	92
7.3	Research Limitation and Future Works .....	95
<b>REFERENCES .....</b>		<b>96</b>

## LIST OF TABLES

<u>Table No.</u>		<u>Page No.</u>
4.1	Details of imbalanced class dataset	36
4.2	Binomial data distribution	37
4.3	Multiclass data distribution	38
6.1	Confusion Matrix results of Algorithms	64
6.2	Accuracy Rate of Algorithms	66
6.3	Weighted mean Recall of Algorithms	70
6.4	Weighted mean Precision of Algorithms	74
6.5	F1 Score of Algorithms	77
6.6	AUC of Algorithms	79
6.7	Yeast Protein Localization data distribution	83
6.8	Ecoli Protein Localization data distribution	84
6.9	The Magnitude of Difference Indicated by A Test Score	90
7.1	Research objective achievements	94

## LIST OF FIGURES

<u>Figure No.</u>		<u>Page No.</u>
2.1	An example of a neural network with two hidden layers (Amato et al., 2013).	12
2.2	SVM mapping nonlinear problem to linear using optimum hyperplane (Ren, 2012).	13
2.3	A schematic design of an RBM architecture (Lopes et al., 2012).	16
2.4	A stacked RBM or known as DBN (Hinton, 2007). The network receives of inputs from the bottom and produces outputs at the top.	16
2.5	An example of a DNN architecture (Richardson et al., 2015).	21
2.6	An example of a CNN architecture (Mrazova & Kukacka, 2012)	23
2.7	An illlustration of GA flow (Assodiky et al., 2017).	26
3.1	Relation between methodology phases, research objectives and research questions	29
4.1	Representation of categories and attributes of datasets	35
4.2	Representation of a confusion matrix	41
4.3	Example of AUC plot.	45
5.1	Flowchart of DBN	54
5.2	Flowchart of Evolutionary DBN with bootstrap sampling	61

# CHAPTER 1

## INTRODUCTION

### 1.1 STUDY BACKGROUND

Imbalanced data classification is a classic yet relevant challenge in machine learning. It deters the optimal prediction of the dataset, which can affect negatively to decision making. When data instances are too expensive to be collected or the data are simply scarce, it emanates data disparity between the categorical classes.

The class with the most instances is called the majority class, while the class with the least instances is called the minority class. Imbalanced data problems usually occurs because the data is unavailable and the attempt to retain it is expensive (Berry et al., 2012). Since machine learning rely on training data to analyse and predict the outcome of the models, it find difficulties when dealing with imbalanced class datasets.

Common outcomes from this issue are misclassification and fluctuating error rates. This pose an issue when the minority class is the sought after prediction as these scenarios does not happen on regular basis. Since imbalanced class problem is a frequent obstacle, there has been many approaches formulated to encounter such complication. Imbalanced class problem can either be solved using data sampling method, algorithm modelling method or both.

Data sampling method involves in tweaking the data itself such as undersampling and oversampling. Undersampling is the process of removing selected instances from the majority class in the dataset. Undersampling might dispose certain instances that can be important for an algorithm model to learn from (Y. Liu et al., 2010). Oversampling is the process of duplicating the instances from the minority class so that it

has the same amount as the instances in the dataset. However, this method can produce a negative effect commonly overfitting (Y. Liu et al., 2010).

Another approach to overcome the repercussions of imbalanced class data is by algorithm modelling. This includes modelling an algorithm hybrid that solves the issues of specific data input. Deep learning algorithms has gained many interests as a state-of-the-art machine learning approaches. Deep learning algorithms has proven to able produce high accuracy results as well as extracting high level abstraction of many domains(Wang et al., 2012; Mohamed, Dahl, & Hinton, 2012; Le & Provost, 2013). Some examples of deep learning algorithms are deep belief network (DBN), deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN) and convolutional deep belief network (CDBN). DBN is a stacked of neurons made up of restricted Boltzmann machine (RBM). The network is fine tuned using backpropagation neural network (BPNN) approach. DBN is widely researched algorithm and is proven to produce good results in high abstract domains such as emotion recognition (Le & Provost, 2013) and acoustic modelling (Mohamed, Hinton, & Penn, 2012).

Evolutionary algorithm (EA) acts as an optimization algorithm. It is inspired by biological evolution process. It involves the process of selection, crossover and recombination that allows improvement in selected aspects of an algorithm. There many types of EA, such as genetic algorithm (GA), particle swarm optimization (PSO)(Y. Zhang et al., 2014) and whale optimization algorithm (WOA)(Mirjalilia & Lewis, 2016). GA is a commonly used EA because it yields good results when utilized as optimization algorithm.

## 1.2 PROBLEM STATEMENT

Imbalanced class can affect negatively in decision making process by providing poor results due to misclassification and fluctuating error rates (Weiss & Provost, 2001). Common methods used to overcome such problem is by data sampling or algorithm modelling (Zhai et al., 2015). Data sampling method is commonly used and is useful when it comes to handling imbalanced class issue because it deals the problem directly. The basic approach that is frequently used is undersampling, oversampling or a hybrid of both. However, the issue with undersampling is that it is possible to get rid of crucial data needed for prediction (H. Han et al., 2005), while the issue with oversampling is that it causes overfitting in learning (Y. Liu et al., 2010). Nevertheless, implementing a data sampling method is still sought after since it can minimizes the negative effects of imbalanced class problems because it deals with the data directly.

Another plausible solution for imbalanced class problem is by algorithm modelling. Deep learning has shown promising results in many domains, especially the ones that require high level abstraction and has complex data features, such as image processing, emotion detection and handwriting recognition (Wang et al., 2012; Mohamed, Dahl, & Hinton, 2012; Le & Provost, 2013). An example of deep learning algorithms is deep belief network (DBN). DBN can learn from complex feature input such as emotion recognition (Le & Provost, 2013) and acoustic modelling (Mohamed, Hinton, & Penn, 2012). Therefore, it can learn the features from an imbalanced class dataset and classify it correctly. Despite the promising performance of DBN in various fields, the algorithm is generally computationally expensive and unable to achieve competent result when learning from inadequate amount of data (Le & Provost, 2013; Mohamed, Hinton, & Penn, 2012).

### **1.3 RESEARCH HYPOTHESES**

Below is the hypothesis of the study:

H<sub>1</sub>: The Evolutionary DBN with bootstrap sampling has shown high performance in terms of performance metrics and statistical analysis as compared to other deep learning and machine learning algorithms when handling imbalanced class datasets.

### **1.4 RESEARCH OBJECTIVES**

This study embarks on the following objectives:

1. To study the effects of DBN algorithm with imbalanced class datasets.
2. To propose an optimized DBN model using an evolutionary algorithm with bootstrap sampling for imbalanced class datasets.
3. To compare the classification performance of optimized DBN algorithm with other machine learning algorithms for imbalanced class datasets.
4. To analyze the performance of optimized DBN using statistical techniques.

### **1.5 RESEARCH QUESTIONS**

This research aims to answer the following questions:

1. How does the imbalanced class datasets affect the performance of DBN algorithm?
2. How to optimize the DBN algorithm?
3. Does the implementation of an evolutionary algorithm and bootstrap sampling on DBN increases the performance when imbalanced class datasets are used?



## **1.6 SIGNIFICANCE AND CONTRIBUTION**

The major contribution of this research is to propose an Evolutionary DBN with bootstrap sampling for imbalanced class problems. This optimized DBN algorithm is helpful for predicting imbalanced class datasets of binomial category.

## **1.7 THESIS STRUCTURE**

This thesis is organized into 6 chapters. Chapter 2 presents the literature review studied for this thesis. The literature review section studies the background of imbalanced class challenges in datasets and deep learning algorithms. Machine learning and evolutionary algorithms are also reviewed for comparison. Chapter 3 provides the research methodology of this thesis. It describes the steps taken to introduce an optimized DBN for imbalanced class data classification.

In Chapter 4, the experimental setup of the research is explained in details. The imbalanced class datasets, algorithms, performance metrics and statistical tests utilized for the experiment is elucidated. Chapter 5 presents the results and performance of Evolutionary DBN with bootstrap sampling and compares it with other deep learning and machine learning algorithms. Finally, Chapter 6 is dedicated to the conclusion of the experiment and future works recommendation based on the result analysis.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

This section presents the literature review done for imbalanced class problem, machine learning, deep learning and evolutionary algorithms. Section 2.2 describes imbalanced class affair in datasets and the usual techniques employed to overcome or minimize its negative effects towards algorithm prediction result. Sections 2.4 and 2.5 explains the details of machine learning and deep learning algorithms in general and their involvement with imbalanced class datasets. Section 2.6 elucidates evolutionary algorithm and its examples when dealing imbalanced class datasets.

#### **2.2 IMBALANCED CLASS DATASET**

Imbalanced class ordeal in a dataset is a common classification task problem. According to Hensman & Masko (2015) and Yan et al. (2015), imbalanced class refers to the *"disparity of data dispensation between the classes"*. The distribution of instances between the classes are not balanced. This affects the performance of an algorithm when doing prediction of the classes (C. Zhang et al., 2018). The class that has more training values is called the majority class and the class that has the least or most missing data values are called the minority class (Swersky et al., 2010).

Minority data class is a realistic problem that the real-world situation faced (C. Zhang et al., 2018) because most of the time even for an important dataset such as cancer detection (N. V. Chawla et al., 2004) and bank fraud (Awoyemi et al., 2018), the data instances are scarce. It can be expensive if the new data needs labelling (Berry

et al., 2012). Unfortunately, most of the algorithms that showed stable and promising performance when using balanced data in classification tasks displayed conflicting outcome when imbalanced class dataset is used (Ghahabi & Hernando, 2014). Prediction of minority class is presumed to have a higher error rate compared to the majority class and its test examples are often wrongly classified as well (Weiss & Provost, 2001). Imbalanced data distribution among the classes causes deficient classification models (C. Zhang et al., 2018). The algorithm that performs on balanced dataset will not perform as good when using an imbalanced dataset (C. Zhang & Tan, 2016), regardless how good the model is.

In a study done by Yan et al. (2015), an imbalanced class dataset in multimedia format is implemented as the input for CNN. The dataset is a TRECVID dataset, which means it is in the form of video. The outcome shows that the error rate fluctuate unlike when using a balanced dataset, the error rate of the algorithm decrease steadily.

There are a few commonly used methods utilized to tackle the challenges of imbalanced class dataset. The first method is using machine learning algorithm and model hybrids according to the input types (W. Liu & Chaw, 2011; Zhai et al., 2015). Another method is by data preprocessing of the imbalanced dataset itself (Zhai et al., 2015). Both data sampling and machine learning approaches are explained in Section 2.3 and Section 2.4 respectively.

### **2.3 DATA SAMPLING METHOD**

Data sampling is a commonly used method to rebalance the data instances (W. Liu & Chaw, 2011). This way, the negative effects of imbalanced class data is handled by directly manipulating the amount of instances in the dataset. In this section, the approaches that are discussed are oversampling, undersampling and bootstrap sampling.

Fernandez et al. (2011) highlights that this method can liberate the performance of an algorithm to be affected.

### **2.3.1 Oversampling**

Oversampling is when the minority class in the dataset are duplicated until it has the same amount or as many as the majority classes in the dataset (Y. Liu et al., 2010; Ganganwar, 2012; W. Liu & Chaw, 2011). Even though it seem ideal for training algorithm because the dataset will have more instances, the disadvantage of an oversampling is that it can cause the classifier the problem of overfitting because the algorithm will learn the redundant values all over again (Y. Liu et al., 2010). "Synthetic Minority Oversampling Technique" (SMOTE) is a common algorithm to optimize the use of oversampling technique (Y. Liu et al., 2010; Fernandez et al., 2011; N. Chawla et al., 2002) . However, there are many variation of SMOTE to suit each dataset (W. Liu & Chaw, 2011). Hensman & Masko (2015) performed an oversampling technique to minimize the effects on convolutional neural network (CNN)

### **2.3.2 Undersampling**

Undersampling is when the majority class in the dataset is downsized, which means the values are removed in order to have the similar amount as the minority data class (Y. Liu et al., 2010; Ganganwar, 2012; Fernandez et al., 2011). If the minority class is desired, the majority class of the dataset is probably unaffected considering the algorithm can focus on the balanced amount of instances instead. However, the downside of an under-sample is that there is a possibility that the important values are taken away from the majority class which can result in an inaccuracy (Y. Liu et al., 2010; H. Han et al., 2005; C. Zhang & Tan, 2016). According to Y. Liu et al. (2010), undersampling is considered

superior to oversampling in terms of avoiding overfitting.

### **2.3.3 Bootstrap Sampling**

Bootstrap sampling is when a small sample is derived from its original sample iteratively (Yan et al., 2015; Rosca, 2014). This method basically reuse its training samples and this is a suitable technique to avoid data redundancy as well as data disposal. Megumi et al. (2015) conducted a neuroscience experiment involving fMRI neurofeedback. Bootstrap sampling was utilized as a method to assess the experiment's difference in correlation between the neurofeedback and other networks.

Bootstrapping sampling is a frequently adopted technique implemented to improve performance of deep learning algorithms with imbalanced class data (Y. Liu et al., 2010; Berry et al., 2012). Yan et al. (2015) implemented convolutional neural network (CNN) to classify an imbalanced multimedia dataset. Bootstrap sampling method is integrated with the algorithm to minimize its fluctuating error rate. The experiment yielded high F1-score as compared to another framework proposed by Tokyo Institute of Technology (TiTech).

In another literature, Berry et al. (2012) implemented bootstrap sampling as a method to improve both computational time and accuracy rate after training the imbalanced and unlabeled data using deep belief network (DBN). The result is recorded to have 41% decrease of error rate that needs human intervention as compared to no bootstrapping implementation. Z. Sun et al. (2018) predicts wind speed and wind power using deep belief network and optimized random forest. The experiment has inconsistent amount of data because some data are simply unavailable. Therefore, the experiment employed bootstrap sampling as an approach to resample the training data to improve the performance of their model.

## 2.4 MACHINE LEARNING ALGORITHM

Machine learning is a field in Artificial Intelligence (AI). It focuses on algorithms that are prototyped from natural intelligence such as the human brain or animal interactions so that it has the ability to learn. Machine learning algorithms are categorised into supervised learning, unsupervised learning and reinforcement learning. Supervised learning is when the input data is labeled and the output needed is classification, recognition or prediction. Unsupervised learning is when the input data available are not labeled and the output often involves clustering.

A few examples of supervised learning algorithms are artificial neural networks (ANN), support vector machine (SVM), naive Bayes, decision trees and k-Nearest Neighbour (k-NN). Although machine learning algorithms are essentially stable and have strong mathematical and statistical basis, it is not robust when applied to different domains since it lacks the domain knowledge and data processing. There are few machine learning algorithms examples that are used to tackle imbalanced dataset classification. In this section, a few examples of the algorithms and the result towards the imbalanced dataset classification will be explained.

The author Y. Liu et al. (2010) used support vector machine (SVM) as the main algorithm and showed that the effect of data disparity results in a *"high false negative rate"*. Zou et al. (2008) applied a Genetic algorithm (GA) sampling to an imbalanced protein data for prediction and continue classifying it with SVM. Another paper, W. Liu & Chaw (2011) modified k-nearest neighbours (kNN) algorithms to counter the effect of imbalanced class dataset to the algorithm.

This section will present the related work conducted in respect to imbalanced dataset classification using ANN and SVM. Relevant works of applying each algorithm

as a method of countering the problems of imbalanced data is presented in each sections.

#### **2.4.1 Backpropagation Neural Network Algorithm**

Artificial neural networks (ANN) are modelled after the human brain networks (D. Zhang & Xu, 2014; Amato et al., 2013). It is widely known used for supervised learning and recognising patterns from input data set by weight adjustments (D. Zhang & Xu, 2014). Common examples of ANN are feed forward, radial basis function (RBF) and Kohonen's Self Organizing Map, to name a few. ANN's ability to scrutinize nonlinear data and to design complex models has allow it to be applied in studies of different fields (Amato et al., 2013; Drew & Monson, 2000).

The most common neural network algorithm used is the backpropagation neural network (BPNN). BPNN has three layers, which consists of an input layer, a hidden layer and an output layer (D. Zhang & Xu, 2014). The layers are made up of interconnected nodes by a weighted association and the number of nodes of the layers depend on the problem domain (Amato et al., 2013). The input layer will accept the data for training or testing and pass the weights to the connected hidden layer. Hidden layer can be one or more and it will continue calculating the weights it received and send it to the output layer where the result is produced. BPNN compares its real output and target output and adjust its weight according to the error and propagates back to its network. However, BPNN is commonly known to have the problem of overfitting when learning (Lanbouri & Achchab, 2015).

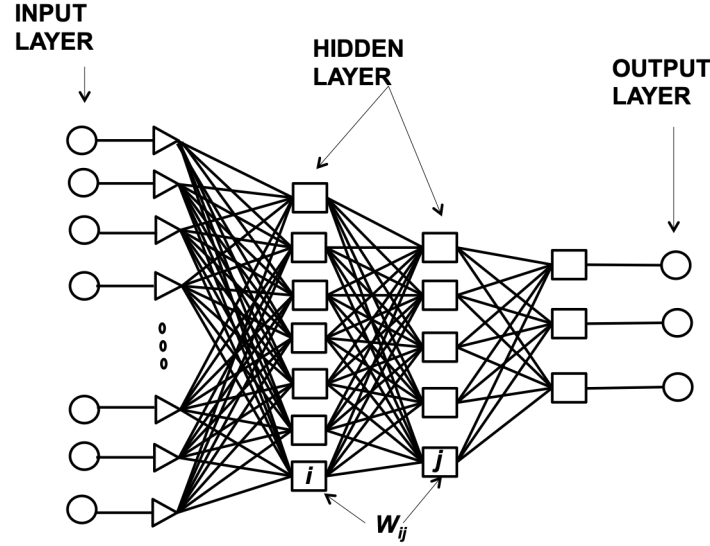


Figure 2.1: An example of a neural network with two hidden layers (Amato et al., 2013).

Arora et al. (2010) implemented back propagation neural network classifier to categorize Devnagari handwritten classes and compared its performance with SVM using the same handwritten data set. The experiment result for BPNN performance is 90.44% for testing data set accuracy. Another work by J.Pradeep et al. (2011) proposed a diagonal based feature extraction and used a "feed forward backpropagation" neural network to classify the data based on the new feature extraction. The experiment achieved 96.52% with 54 features and 97.84% with 69 features accuracy rate.

In tackling imbalanced data, Cao et al. (2013) presented a cost sensitive back-propagation neural network for a multiclass imbalanced data, as opposed to the "limited" binary class imbalanced data. D. Zhang & Xu (2014) implemented BPNN as a method to credit scoring. The dataset is from "The German Credit Dataset" and is imbalanced. The dataset is meant to separate between eligible credit applicants. Other than that, BPNN is also employed to predict protein disorder and manage to get accuracy rate of 91.00% on average of 4 BPNN models (Oh, 2013). The dataset is imbalanced and is procured from "Korea Institute for Advanced Study"



### 2.4.2 Support Vector Machine Algorithm

According to Arora et al. (2010), SVM can be defined as a "binary classifier", where the outcome will be divided into two groups based on the optimum hyperplane.

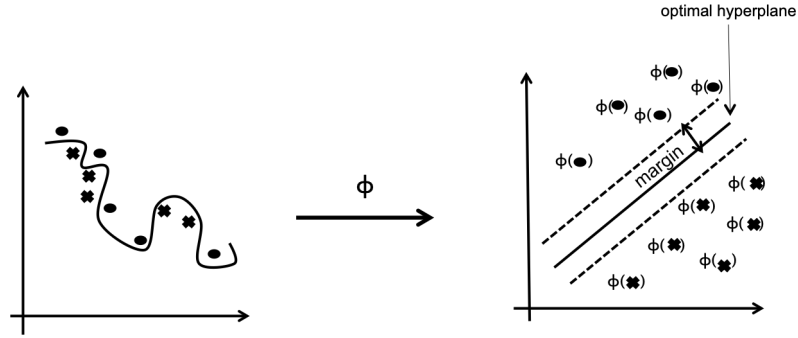


Figure 2.2: SVM mapping nonlinear problem to linear using optimum hyperplane (Ren, 2012).

Niu & Suen (2011) implemented a hybrid of SVM and CNN for classifying MNIST handwritten digits dataset. Feature extraction is done using CNN and SVM acts as a "recognizer". (Arora et al., 2010) compared the performance of SVM and ANN using the Devnagari handwritten recognition problem. SVM performance in the experiment achieved 92.38% for testing accuracy.

In countering the imbalanced data classification problem using SVM, its weight and activation function are manipulated in order to increase the classification accuracy (Hwang et al., 2011). Tang et al. (2009) stated that SVM outperforms other conventional classifiers when a moderate imbalanced data is used. Even so, when a high imbalanced data is used instead, SVM classifier can still produce a biased result. Most works using SVM to counter imbalanced data only focused on the performance and not efficiency, hence, SVM can be a slow classifier (Tang et al., 2009). However, Zou et al. (2008) stated that SVM could not perform imbalanced data classification successfully based on

the works of Cristianini & Shawe-Taylor (2000).

Koturwar et al. (2015) stated that SVM has the ability to balance massive data correctly. Feature extraction using SVM is good as it can be done promptly using SVM kernel instead of a feature extraction process that results to data lost (Koturwar et al., 2015). One of the disadvantages of SVM classifier is its training and execution is very complex caused it to be implemented in mostly small category set problem (Arora et al., 2010). According to Koturwar et al. (2015), large training data makes SVM inefficient and costly as SVM is not scalable to huge size data. When the training data is noisy and imbalanced, it can effect the outcome of SVM due to its high training execution and low generalization error (Koturwar et al., 2015).

## **2.5 DEEP LEARNING ALGORITHM**

Deep learning is a part of machine learning algorithms. It is recently introduced to solve heterogeneous and high dimensionality datasets, such as image and audio data. There are several types of deep learning architectures which are deep neural networks (DNN), recurrent neural networks (RNN), convolutional neural networks (CNN), deep belief networks (DBN) and convolutional deep belief networks (CDBN).

DNN is an intricate layers of artificial neural network with multiple hidden layers of units between the input and output layers. DNNs can model complex non-linear relationships which make it suitable for object detection. An RNN is a network where the neurons are reused to produce output. Hence, the outputs are depending on the previous computations. A CNN is composed of one or more convolutional layers with fully connected layers on top. CNN is commonly used in computer vision and acoustic modelling for automatic speech recognition (ASR). A deep belief network (DBN) is a probabilistic, generative model made up of multiple layers of hidden units. It can be

seen as a composition of simple learning modules that make up each layer. A CDBN, on the other hand, is a combination of CNN and DBN. Its ability to exploit 2D image structures is due to its similarity of architecture with CNN. Whereas, it practices data pre-training that is of DBN. They provide a generic structure, which can be used in many image and signal processing tasks.

Encouraging results have been displayed upon the application of deep learning algorithm in text and handwritten recognition (Y. Liu et al., 2010; Wang et al., 2012), audio classification (Mohamed, Dahl, & Hinton, 2012; Dieleman et al., 2011) and even abstract high-level domains such as emotional recognition (Le & Provost, 2013; C. Huang et al., 2014). However, these are applied for data that are distributed evenly. Not many imbalanced class problems have been solved using a deep learning method. According to Deng (2012), deep learning algorithm is a type of machine learning method and is thriving when applied to pattern classification and feature learning. It has deeper architecture structure compared to previous machine learning algorithms (Deng, 2012). In this thesis, DBN is chosen as the deep learning algorithm implemented to tackle imbalanced dataset classification problem. Literature review of other deep learning algorithms such as DNN and CNN is also included for comparison with DBN.

### **2.5.1 Deep Belief Network Algorithm**

To understand DBN, the concept of Restricted Boltzmann Machine (RBM) must first be explained. The architecture of RBM consists of a bidirectional connection between hidden layers and visible layers. This feature allows the weight to be connected exclusively and allows deeper extraction between the neurons. RBM is a probabilistic model (Dieleman et al., 2011) and a two-layer, bipartite, undirected graphical model with a set of binary hidden random variables (units)  $h$  of dimension  $K$ , a set of (binary or real-

valued) visible random variables (units)  $v$  of dimension  $D$ , and symmetric connections between these two layers represented by a weight matrix ( $W \in \mathbb{R}^{Dk}$ ) (Honglak Lee & Ng, 2011). This means that RBM units are assigned statistically random with values 0 or 1. According to Zheng et al. (2014), the output layer of the lower level layer in the RBM will be the input layer to its higher level layer in a bottom-up manner. This allows the pre-training of weight occurs within the DBN and thus increase its accuracy level.

Le & Provost (2013) explained that there are two common types of RBM. The first one is Bernoulli RBM, where the visible and hidden layers are binary. The second one is Gaussian RBM, which has real number values for its visible and hidden units.

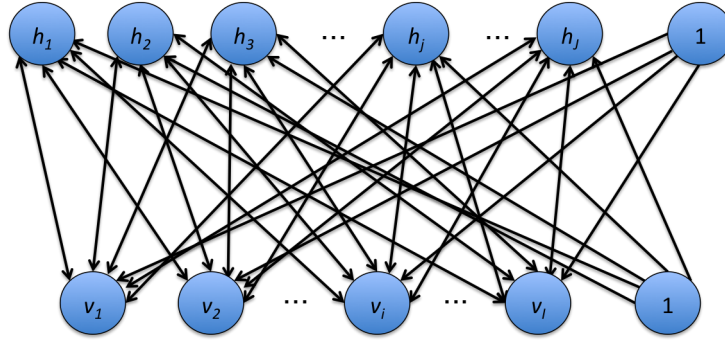


Figure 2.3: A schematic design of an RBM architecture (Lopes et al., 2012).

The figure above presents the schematic design of an RBM architecture. RBM is made up of a stochastic visible units and stochastic hidden units that are connected to each other (Mohamed et al., 2010).

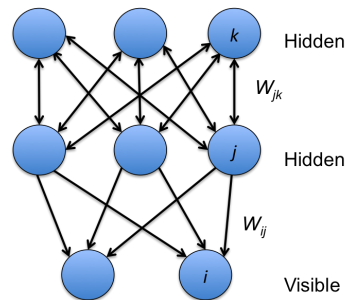


Figure 2.4: A stacked RBM or known as DBN (Hinton, 2007). The network receives of inputs from the bottom and produces outputs at the top.

A deep belief network (DBN) is a probabilistic, generative model made up of multiple layers of binary latent units (Mohamed, Hinton, & Penn, 2012). As shown in the figure above, DBN is a composition of simple learning modules that make up each layer in the bottom-up fashion. DBN incrementally made up of RBM-trained layer-by-layer in greedy method (Terusaki & Stigliani, 2014).

Compared to a neural network's ability to represent input vector simultaneously and combine both discrete and continuous features of the inputs (Mohamed, Hinton, & Penn, 2012), DBN's generative pre-training adds a higher level of feature abstraction of the input in the network. Neural network layers are exponentially dense (Mohamed, Hinton, & Penn, 2012) but the deepness of DBN allows low level feature abstraction handled by the lower layers and the high level or nonlinear feature abstraction handled by the higher layers of the network.

However, such feature results DBN to be computationally expensive and time-consuming because the number of layers DBN needs to go through is a lot. According to an experiment done by Le & Provost (2013), training a DBN is expensive in terms of computation because pre-training took 11 minutes per epoch and fine tuning takes up 10 minutes per epoch. In another research, the feature extraction using DBN for a speech emotion recognition, the time cost 136 hours and it is longer compared to other methods used for feature extraction in a dataset (C. Huang et al., 2014).

Generative pre-training of weights in DBN are important to augment the possibilities of the input through layers from below (Mohamed, Hinton, & Penn, 2012) to make it more accurate, but this results in an expensive computation of the network. However, according to Hinton (2002), this problem can be improved by applying a training method named "*contrastive divergence*" on every layer. Another common problem with DBN is the parameter setting (Jo & Lee, 2015). There is a lot of effort to combine

the settings of DBN to finally get the best performance. Although DBN shows it can learn from imbalanced class dataset better than CNN, the time taken for training is long (Amri et al., 2017).

DBN showed good result for high level abstraction domain datasets. DBN is used in emotions recognition (Le & Provost, 2013) by learning high-level features in the dataset. Terusaki & Stigliani (2014) also used DBN to do an emotion detection by using a facial expression dataset. Face verification also use DBN, while combining the algorithm with CNN. The hybrid algorithm aims to achieve robustness in verifying similarities of different faces (Y. Sun et al., 2013). DBN is also used to model natural images (Aurelio et al., 2010) by learning multiple layers of unlabelled data. Acoustic modelling is a thriving domain that uses DBN due to its flexibility by having non-linear hidden layers and being generatively pre-trained which makes it more discriminative (Mohamed, Hinton, & Penn, 2012).

Ma & Wu (2014) used DBN to do a representation learning method for an online handwritten Tibetan characters recognition. DBN is also used to classify a set of Bangla numerals and achieved 90.27% accuracy rate (Sazal et al., 2013). Al-Fatlawi et al. (2016) employed DBN as a method to perform feature extraction in order to diagnose Parkinson's disease. The model is fine-tuned using backpropagation neural network (BPNN) and achieved overall 94% accuracy testing performance. Jang et al. (2017) performed a classification on fMRI volumes using DNN, but the model is pretrained with DBN.

Another experiment classify SAR images by proposing a discriminant DBN (DisDBN) that features DBN and ensemble learning (Zhao et al., 2017). Yu et al. (2015) proposed to perform credit risk assessment by using multi-stage DBN with ensemble learning of extreme learning machine (ELM). ELM is used as a forecasting stage while

DBN is executed in the final stage of output classification. Jiang et al. (2016) tackles a text classification problem by employing DBN for feature extraction and classify using softmax regression. Xie et al. (2010) used a combination of DBN and sparse coding and called it Sparse Deep Belief Nets to classify MNIST handwritten digit dataset.

Walid & Lasfar (2014) had implemented sparse DBN for ICDAR 2013 handwritten dataset recognition. Hyperspectral data classification yields better result when the network consists of DBN for feature extraction and logistic regression (LR) for classification (Y. Chen, 2015). In a speech recognition problem, a 5-layer DBN is used to extract the speech emotion features of the data. SVM was used as the classification phase. The result achieved by the author was increased by 7% compared to when using normal classifier and only SVM. Huang et al. (2014).

In time series forecasting domain using DBN, Kuremoto et al. (2014) implemented particle swarm optimization (PSO) to determine the visible and hidden layers and the learning rates of RBM and BPNN used in the DBN model. The proposed method manage to work better when tested on different chaotic time series data. Its root mean squared error (RMSE) value dropped for both Lorenz chaos and logistic map.

A financial distress prediction using real life dataset implement a DBN hybrid to perform the prediction (Lanbouri & Achchab, 2015). The dataset is classified into two categories, distressed and non-distressed for "Micro-business" and "Small and Medium Business" (SMB), and the ratio is stated to be imbalanced. RBM feature of DBN is used as a pre-training for the training data, and SVM is employed for classification phase. The result is 76.8% accurate as compared to 62.1% by ANN.

Berry et al. (2012) utilized DBN to cater to an imbalanced unlabeled data. The dataset consists of ultrasound images of tongue when a subject is performing human speech. Imbalanced in speech data is simply unavoidable because some images will

look similar than the rest as stated in Zipf's Law. Bootstrapping is incorporated into DBN to reduce computation time. The method proves to improve the accuracy and reduced time taken for labelling the data.

Kuang & He (2014) attempts to classify fMRI datasets using DBN to predict whether a patient has ADHD or not. The ADHD dataset is imbalanced, and its effect on DBN is low accuracy rate. Therefore, the dataset went through preprocessing methods and this approach saw an increase of accuracy rate using DBN.

A balanced dataset would have a fairly equal misclassification costs by conventional algorithm, but due to the disparity of data distribution between the classes has made this assumption to cause distortion in the results. Therefore, C. Zhang & Tan (2016) proposed a cost-sensitive DBN to handle imbalanced class dataset by assuming the misclassification costs between the classes are not equivalent. Evolutionary algorithm is also utilized as a part of the proposed method to find optimal solution of cost in the cost matrix.

### **2.5.2 Deep Neural Network Algorithm**

As the name suggests, deep neural network (DNN) is an intricate layer of feedforward neural network (FFNN)(Richardson et al., 2015). A typical FFNN is a multi-layer perceptron (MLP) that is made up of an input layer, a hidden layer and an output layer. DNN, on the other hand, possess an input layer, more than one hidden layers and an output layer(Richardson et al., 2015; K. Han et al., 2014a). Therefore, training a DNN is invariable from training basic MLP (Richardson et al., 2015).



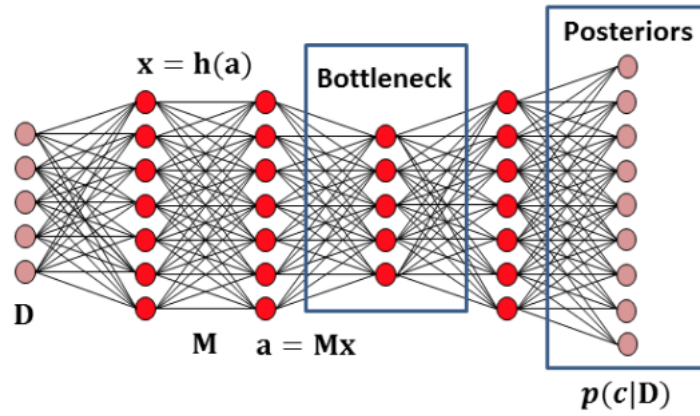


Figure 2.5: An example of a DNN architecture (Richardson et al., 2015).

Domains that has applied DNN algorithm as a classifier includes acoustic models (Richardson et al., 2015) and speech emotion recognition (K. Han et al., 2014a). Other than, DNN is also used for mathematical based problems such as compressed learning (Adler et al., 2016) and semantic segmentation (Hong et al., 2015). D. Cirean, Meier, & Schmidhuber (2012) used multi-column DNN to classify MNIST dataset and receive an error rate of 23%.

Jang et al. (2017) performed DBN as a pre-training for DNN to classify fMRI volumes. D. Cirean, Meier, Masci, & Schmidhuber (2012) used DNN to classify traffic sign. The traffic sign dataset is a part of image classification problem. DNN manage to successfully recognize and classify with the rate of 98.52% in the experiment. However, when DNN is added with convolutional layers and introduced as "Multi-Column", the accuracy rate increases to 99.46%.

DNN is also implemented to recognize emotion through speech dataset (K. Han et al., 2014b). The authors combined DNN to predict emotion based on segment-level, and extreme machine learning (ELM) on utterance-level. Kaur et al. (2018) classified speech recognition using DNN but implemented genetic algorithm (GA) as feature optimization for the dataset. The proposed classifier managed to achieve accuracy rate of

94%. DNN showed an accuracy rate of more than 90% in the phonetic features of basic speech domain as compared to shallow MLP (Siniscalchi et al., 2013). J.-T. Huang et al. (2013) investigates the whether a DNN-based classifier that managed to have good performance when learning for large vocabulary speech recognition (LVSR) tasks can achieve the similar performance when other European languages are used. The training set consists of multiple European languages such as German, French, Italian and Spanish and each dataset has speech data ranges from 195-hour to 63-hour. The DNN-based classifier with a slight modification of shared hidden layer manage to achieve good performance with 3-5% error reduction. X. Zhang et al. (2014) also implemented DNN-based classifier to a LVSR task and incorporated the algorithm with Max-out nonlinearities with different activation functions. These experiments describe the ability of DNN to learn from complex features and domain.

Although DNN classifier and training has a superiority compared to a straightforward MLP, the time taken for DNN's computation is very slow(Sainath et al., 2013). Deng et al. (2013) mentions that even though DNN outperforms shallow nets such as Gaussian Mixture Model (GMM) and neural nets to perform acoustic modelling, DNN training time is slow. DNN is also affected by imbalanced datasets. Khan et al. (2016) proposed a cost-sensitive DNN to minimize the computation loss when DNN is used on a multiclass imbalanced dataset.

### **2.5.3 Convolutional Neural Network Algorithm**

A convolutional neural network (CNN) consists of one or more convolutional layers (Hensman & Masko, 2015; Yan et al., 2015; Abdel-Hamid et al., 2013), alternating with subsampling layers and by the end of the network, optionally, a fully connected MLP (Hensman & Masko, 2015). A CNN architecture is consisted of one or more convolu-

tional, pooling and a fully connected layers on top (Yan et al., 2015). The convolutional layers are responsible for feature extraction and is called a feature map (Hensman & Masko, 2015; Yan et al., 2015; Abdel-Hamid et al., 2013) or a feature detection in other literature (Matsugu et al., 2003).

After convolutional layer, it is often paired up with a pooling layer that will perform a pooling function based on the inputs it received from the previous convolutional layer (Hensman & Masko, 2015; Yan et al., 2015; Y. Liu et al., 2010; Fernandez et al., 2011). The pooling layer is also known as a subsampling layer, and it will alternate with a convolutional layer because it computes the statistics of the convolutional layer. The pooling layer will perform pooling functions and is called min-pooling, max-pooling layers or etc. according to its context of problem-solving. The pooling function will "*downsample*" the input it received from its convolutional layer (Yan et al., 2015). This process continues until the end of the network.

At the end of the series of alternation, a fully connected MLP will be added. It works as a classification module for the network (Hensman & Masko, 2015). This layer will receive all neurons from its previous layers whether they are convolutional or pooling and connect them with its own neurons (Yan et al., 2015).

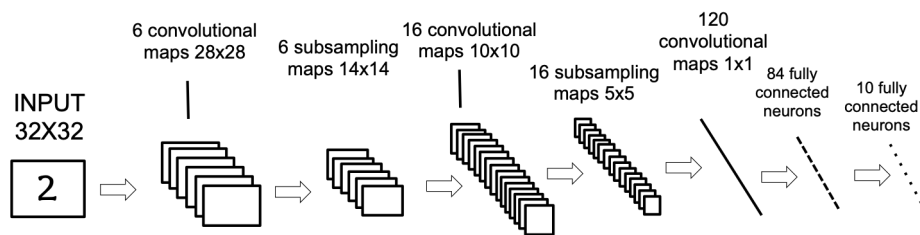


Figure 2.6: An example of a CNN architecture (Mrazova & Kukacka, 2012)

However, the implementation of convolutional and subsampling layers in a CNN and the method used for the network training differs in every CNN (D. C. Ciresan et al., 2011). The approach depends on the context of the problem that is required to

solve. Zhao et al. (2016) combined the convolutional and pooling features of CNN with syntax word embedding in order to extract context from biomedical dataset. The dataset aims to search the drugs combination prescribed and whether or not the combination is dangerous to patients and is stated to be imbalanced.

Yan et al. (2015) implemented a CNN-based algorithm integrated with bootstrap sampling to boost the classification performance of an imbalanced TRECVID 2011 multimedia dataset. This approach manage to gain higher F-score and lower training time. Acar (2013) used music video clips labeled with each genre and is imbalanced. CNN was incorporated to learn mid-level representation from the music video clips and SVM is used for classification. The approach manage to get an increase of 20% as compared to low-level feature learning. However, Acar (2013) mentioned that the classifier still has low accuracy rate due to data imbalanced.

#### **2.5.4 Findings in Deep Learning Literature Study**

In summary, it can be said that DBN has an advantage over DNN and CNN when dealing with imbalanced class datasets. CNN is mostly implemented in visual recognition domain. Even though DNN has advantages over shallow neural networks to overcome the problem domain, its main issue would be its computation time. Considering this thesis aims to see the effects on performance of an algorithm when handling imbalanced class dataset from multiple domains and structures and proposing some modifications to improve its performance, DBN is chosen as the main focus in the study.

### **2.6 EVOLUTIONARY ALGORITHMS**

Evolutionary algorithms (EA) is inspired by the biological process of evolution. The main process EA go through is election, reproduction, mutation and recombination that

allows the solutions keep improving over time(Bliss et al., 2014). EA is typically used to optimize an algorithm because of its capability to select the best solution and keeps recombining until the best fit is achieved. EA is very useful in to solve issues that requires dynamic solutions (Bliss et al., 2014; Khandelwal et al., 2018). This makes EA a versatile algorithm that thrive cross domains. EA is successfully implemented in various domains and used to solve many problems.

Bliss et al. (2014) used EA as a part of the solution to predict future links in dynamic social networks by weight optimization. In another work, Bliss et al. (2014) proposed a tree based grammar representation to identify dynamical systems. However, the proposed solution derived an additional issue known as multi-objective optimization problem. Therefore, the authors employed EA based approach called the Genetic Programming (GP) to overcome the issue by optimizing parameter selection.

Some examples of EA are nature inspired algorithms, such as whale optimization algorithm (WOA), particle swarm optimization (PSO) and genetic algorithm (GA). GA is the most popular optimization algorithm because of its feature of selecting the best solution based on a fitness calculation. This feature allows GA to perform well in optimizing other algorithms in solving classification problems.

### **2.6.1 Genetic Algorithm**

Genetic algorithm (GA) is a heuristic search algorithm that models from the biological evolution (Ganguly & Samajpati, 2015; Qiu et al., 2015). It is introduced by John Holland in the 1970s. GA mimics the human genetic mutation and selection process (Ghamisi & Benediktsson, 2007). GA is made up of chromosome. A chromosome contains multiple genes, and a collection of chromosomes is called a population (Ghamisi & Benediktsson, 2007; D. Liu et al., 2014; Ganguly & Samajpati, 2015; Qiu et al., 2015).

The objective of GA is to ensure that the next iteration has better chromosomes than its previous ones. Therefore, a selection of fitness functions is used as a yardstick to verify that the process is successful. It has various mapping techniques and fitness measurements (Jamshidi et al., 2015). The crossover and mutation features of GA creates randomness in the population allows the heuristics to avoid from local optima solutions (Haque et al., 2016). D. Liu et al. (2014) states that it is an ideal algorithm to be used in fields such as optimization and forecast. The graphic representation of the flow is shown below.

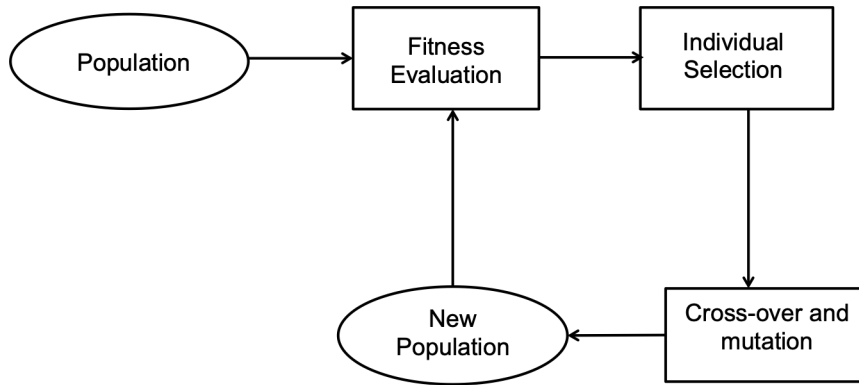


Figure 2.7: An illustration of GA flow (Assodiky et al., 2017).

In latest researches, GA is commonly used to improve or a part of a hybrid algorithm when it comes to prediction and classification. GA has been used as a hybrid with a Particle Swarm Optimization (PSO) to improve a feature selection process of the Indian Pines hyperspectral data set (Ghamisi & Benediktsson, 2007). Jamshidi et al. (2015) has used GA as a part of optimization in removing an element in chemistry domain.

GA is also used to optimize SVM in an application using wavelet transform to forecast short-term wind speed (D. Liu et al., 2014). Other than that, GA is utilized as an optimization for CMP (Qiu et al., 2015). Elhoseny et al. (2015) has employed GA to balance the energy consumption in WSN domain. The data used is heterogeneous.

GA is applied in as a feature selection for a credit risk assessment based on a bank in Croatia (Oreski & Oreski, 2014). Neath et al. (2014) applied GA to obtain the ideal level of performance for the proportional-integral-derivative (PID) controller by tuning its parameter. Assodiky et al. (2017) used GA as a feature selection as a part of a H2O Deep Learning in order to classify ECG data to detect Arrhythmia. Inanlo & Zadeh (2016) applied classification on social networks using GA-based DBN. The network has converged properly and is stable to classify social networks dataset.

For imbalanced data classification task, GA is used efficiently to overcome the common problems through its selective feature. Deshmukh & Akarte (2016) has used GA as an approach to improve SVM for its imbalanced medical data task. Haque et al. (2016) took an approach to use a heterogeneous Ensemble of Classifiers (EoC) in order to overcome imbalanced data problem through generalization. The authors proposed a GA-based technique to appoint the best classifiers that will build a good heterogenous EoC and acquired better result than base classifier and other ensembles. Another method to deal with imbalanced dataset is by using the cost matrix method.

Perry et al. (2015) integrated GA to produce cost matrices that will allow the algorithm to deal with different use-cases of imbalanced data efficiently. GA is known to be robust and a good optimization algorithm (D. Liu et al., 2014). Haque et al. (2016) stated that GA is suitable for tasks that are massive and elaborate because it is less likely to stuck in local optima unlike other heuristics. However, Asadi et al. (2014) claims that GA is computationally expensive when an evaluation function is needed to be executed many times . Therefore, the overall context of the datasets needs to be taken into account if the algorithm aims to be cost sensitive.

## **CHAPTER 3**

### **RESEARCH METHODOLOGY**

#### **3.1 INTRODUCTION**

This chapter presents the research methodology for this thesis. The steps taken are explained under Section 3.1 in details. The research starts with a study of literature review to further understand the problem statement and previous studies involving the field. Then, an experimental setup including imbalanced class datasets and the proposed algorithm is designed. Evaluations and comparisons of the results are done to assess the performance of the proposed algorithm. The output of the experiment is recorded and documented. A summary of the research methodology and its relevance with research objectives and research questions is shown in Figure 3.1.



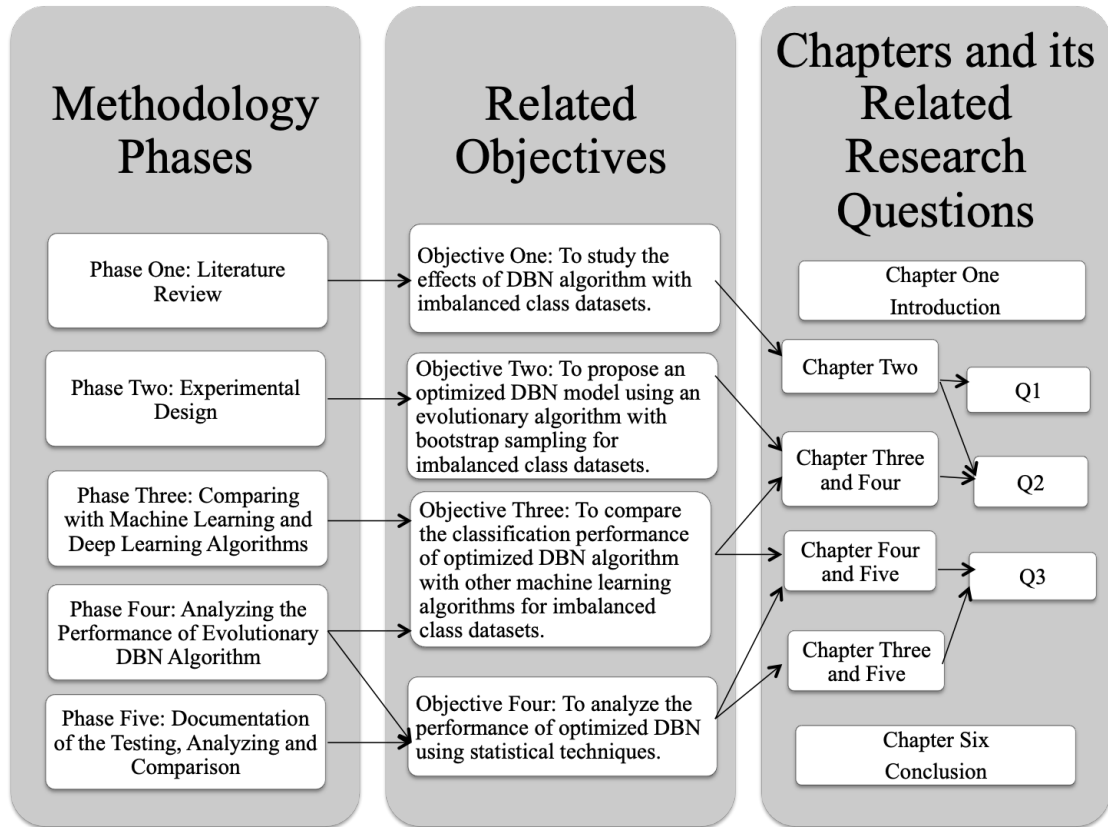


Figure 3.1: Relation between methodology phases, research objectives and research questions

### 3.1.1 Studying the Literature

In the first phase, a study of literature is performed to comprehend the negative effects of imbalanced class dataset on machine learning algorithms and its relevance in real life problems. The inequality distribution of instances in a dataset affects the preciseness of a classification model. Common approaches to overcome the ramifications of imbalanced class datasets are data sampling method or design a stable and robust algorithm that address the problem.

This thesis focuses on utilizing an algorithm to reduce the disadvantageous effects of imbalanced class datasets. Deep belief network (DBN) is generative and made up of stacked Restricted Boltzmann Machine (RBM) in greedy manner. DBN is feasible

in learning from small dataset and manage to solve the vanishing gradient problem that occurs frequently with shallow networks (Terusaki & Stigliani, 2014). Therefore, it is chosen as the main focus in this thesis.

However, this does not guarantee DBN can always overcome the negative effects of imbalanced class datasets. There is no exact amount of data or specific number of layers of DBN that can consistently yield good results when classification task is executed. DBN is also computationally expensive because of its deep architecture.

Literature reviews encompassing other deep learning algorithms, such as deep neural network (DNN) and convolutional neural network (CNN), and selected machine learning algorithms, which are backpropagation neural network (BPNN) and support vector machine (SVM), are also performed to contrast with DBN.

Literature surveys for evolutionary algorithms and sampling method are also investigated to choose the best method to optimize the performance of DBN and its error tolerance when handling imbalanced class datasets.

### **3.1.2 Selecting the Datasets for the Experiment**

In the second phase, imbalanced class datasets are acquired as a part of the experiment setup. In order for the datasets to be a part of the experiment, the classes in the datasets must have unequal distribution of instances between them. The datasets are categorized according to their classification type and their attribute details.

There are 3 categories of datasets which are binomial, nominal and ordinal. A dataset is in binomial category when its output has the chance of two classes. A dataset in nominal category has the chance to be classified in one out of more than two classes. It is also known as a multiclass dataset. Ordinal category is also multiclass but unlike nominal, its classes is in ordered form.

The imbalanced class datasets are categorized according to its attributes characteristic as well. There are 4 types of attributes characteristics, which are binary, numerical and multivariate. A dataset has binary attributes when it is made up of binary values. A dataset with numerical attributes is made up of number values. There are 2 types of numerical attributes which are discrete and continuous. Discrete numerical is when the attributes are made up of integer values, whereas continuous numerical is when the attributes are made up of floating number values. A dataset with multivariate attribute characteristic consists of more than one type of structure.

The datasets are taken from UCI Machine Learning repository (*UCI Machine Learning*, 2018). The details and distribution of the instances are reviewed in Section 4.2 to ensure it is suitable for the experiment. The datasets are also selected based on works performed by C. Zhang et al. (2018), Boughorbel et al. (2017), Weiss & Provost (2001) and López et al. (2013) that involves the issue of imbalanced class datasets.

### **3.1.3 Implementing Deep Belief Network Model for the Imbalanced Class Datasets**

In the third step, each imbalanced class dataset is used as inputs into a deep belief network (DBN) framework. Previous works by Amri et al. (2017) and Amri et al. (2018), DBN shows promising performance when handling imbalanced class dataset. DBN algorithm is translated into Python code. The hidden layer structure, learning rate and activation function for the algorithm is initialized. As each imbalanced class dataset has a different number of attributes and attribute types, the input reader of DBN algorithm is modified to suit the particular imbalanced class dataset. However, the flow of the algorithm and the calculation is unaltered and constant for all imbalanced class dataset.

### **3.1.4 Implementing Evolutionary Algorithm and Bootstrap Sampling for the Deep Belief Network**

After studying and analysing optimization algorithms as mentioned in Section 3.1.1, the best method to apply on DBN is with evolutionary algorithm. Evolutionary algorithm has four main phase. The first phase is initialization, where the population size is initialized. Then, the selection part is commenced, where the members of the population called chromosomes are evaluated using a fitness function.

Fitness function decides if a chromosome is suitable for the next step. Next, crossover and mutation of the selected chromosome is performed. This part will keep iterated and evaluated with fitness function. The selected chromosome is ensured to be better than its previous form. This feature of evolutionary algorithm prevents the algorithm to stuck at local minima. The last phase of evolutionary algorithm is termination. The algorithm will stop itself from generating new version of population after it satisfies a given limit. Bootstrap sampling strategy is also implemented into DBN algorithm to reduce the probability of partiality on data level. Bootstrap sampling is a popular method to be used to counter imbalanced class problem in datasets (Y. Liu et al., 2010; Berry et al., 2012).

### **3.1.5 Analyzing the Performances of Algorithms on the Datasets**

After the optimized algorithm is executed, a series of analysis is carried out:

1. **The Performance Metrics of Algorithms for Each Dataset.** The overall performances of the algorithm for each dataset must be scrutinized to avoid biased perception. Each of the performance metric is taken into account and analysed to provide a comprehensive performance of each algorithm on each dataset. The performance metrics taken into account are accuracy rate, weighted mean recall, weighted mean

precision, F1-score, area under curve (AUC) and confusion matrix. The details of the evaluation metrics are discussed in Section 4.3.1.

**2. The Statistical Test for Each Algorithm.** The result of the data classification is used as parameters for a chosen statistical test. The statistical test will provide the efficacy of the proposed algorithm. The statistical tests used for this research are the Wilcoxon Signed Rank test and Vargha-Delaney A test. The outcome of the tests will determine whether the proposed algorithm performed the best contrast to other algorithms used as comparison. Further description is provided in Section 4.3.2.

### **3.1.6 Comparing the Proposed Algorithm with Other Algorithms**

After the proposed algorithm has completed its task on each dataset, a comparison with other machine learning algorithms is carried out to differentiate its efficiency. The contrast between Evolutionary DBN with bootstrap sampling to other deep learning algorithm such as DBN and DNN is recorded and analyzed. Machine learning algorithms such as backpropagation neural network (BPNN) and support vector machine (SVM) are also collected and assessed to further analyze the performance of the proposed algorithm as done in a work by C. Zhang et al. (2018).

### **3.1.7 Documenting the Results of the Simulation, Evaluation, Comparisons and Analysis**

After previous sections (Section 3.1.1 to 3.1.6) have been done, the results of the testing, the analysing and the comparison are documented. The performance and achievement of the proposed algorithm are compared and discussed in detail. The conclusion of the experiment is presented.

## **CHAPTER 4**

### **EXPERIMENTAL SETUP**

#### **4.1 INTRODUCTION**

In this chapter, the experimental setup for the thesis is explained. Section 4.2 presents the details of imbalanced class datasets used as a part of the experiment. Section 4.3 elucidates the evaluation metrics utilized to assess the performance of the algorithm for each imbalanced class dataset and statistical tests performed to evaluate Evolutionary DBN with bootstrap sampling validity.

#### **4.2 IMBALANCED CLASS DATASET**

This section presents the details of imbalanced class datasets used for the experiment. There are two types of data, namely quantitative and qualitative. Quantitative data provides quantifiable and numerical sample. Quantitative data is divided into two categories, which are discrete and continuous. Discrete data shows the label in a categorical manner such as integer value. Continuous data is when the data is represented in a more precise form such as float value.

Qualitative data provides classes. Qualitative data has three primary types, which are binomial data, nominal data and ordinal data. Binomial data is binary valued. It has two classes. Nominal data is when the instances can be classified into more than two classes. Ordinal data is also when the instances can be classified into more than two classes, but in ordered form. An instance is the single item in a dataset. An attribute is the label for the instance. The characteristic of the attribute is taken into consideration when making classification.

Attributes that has binary characteristic has two states, such as "true" and "false" or "0" and "1". Numerical attribute is when the attributes in the dataset are structured in numbers. As mentioned earlier, numerical sample has two types, which are discrete and continuous. Multivariate attribute occurs when the attributes contain more than one type of structure. For example, a multivariate attributes consist of attributes made of string, integer, float and binary in one dataset.

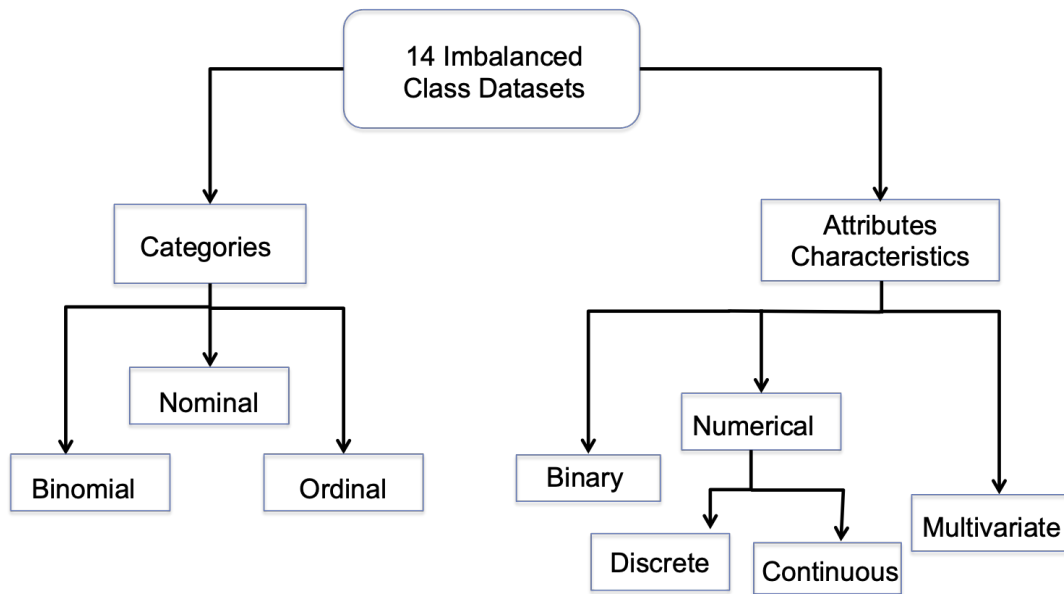


Figure 4.1: Representation of categories and attributes of datasets

The datasets are chosen based on the data disparity of the instances between their classes. The datasets are also chosen based on similar studies using the same datasets conducted by C. Zhang et al. (2018), Boughorbel et al. (2017), Weiss & Provost (2001) and López et al. (2013). The data distribution are displayed according to binomial and multiclass category in Table 4.2 and Table 4.3 respectively.

Table 4.1  
Details of imbalanced class dataset

Category	Attributes Characteristic	Name	Instances	Attributes
Binomial	Binary	Hepatitis	155	20
		SPECT Heart	267	23
	Discrete Numerical	SPECTF Heart	267	45
		Haberman's Survival	306	4
	Continuous Numerical	Parkinson's Disease	197	24
Multivariate	Thoracic Surgery	470	17	
Nominal	Binary	Primary Tumor	339	18
		Zoo	101	18
	Continuous Numerical	Ecoli Protein Localization	336	8
		Yeast Protein Localization	1484	9
Multivariate	Standardized Audiology	200	71	
Ordinal	Discrete Numerical	Contraceptive Method Choice	1473	10
		Dermatology	366	34
	Multivariate	Post-operative	90	9

Table 4.1 presents the information obtained from the imbalanced class datasets. The task that can be done with the datasets is classification task, where the instances are categorized into a class based on their attributes. There is a total of 14 imbalanced class datasets utilized in the experiment. The datasets are varied in their instances and attributes, unless stated.

The number of instances might affect positively towards prediction because the algorithm has many input examples to learn from and adjust the network's weights accordingly. Meanwhile, the number of attributes might affect negatively towards prediction because the algorithm might use too many aspects than needed and at times can make the network learning be inconsistent. Other than that, the category and attributes characteristic could also be an important factor for an algorithm's ability to learn. There is a chance it is easier to classify binomial datasets as compared to nominal and ordinal datasets. Also, it might be easier to learn and classify attributes in numerical form as compared to multivariate form.

In binomial category, the dataset with the most instances is the "Thoracic Surgery" dataset, while "Hepatitis" dataset has the least instances. Other than that, "SPECTF



Heart" dataset has the most number of attributes. "SPECTF Heart" dataset has the same instances number with "SPECT Heart" dataset, but the former is made up of discrete numerical while "SPECT Heart" dataset is made up of binary values for its attributes. "Haberman's Survival" dataset has the least number of attributes. This is to evaluate whether the attributes characteristic affects the algorithm's performance.

In nominal category, the dataset with the most instances is the "Yeast Protein Localization" dataset, while "Zoo" dataset has the least instances. "Standardized Audiology" dataset has the most attributes, while "Ecoli Protein Localization" dataset has the least attributes in the category.

For ordinal category, "Contraceptive Method Choice" dataset has the most instances, whereas "Post-operative" dataset has the least instances. "Dermatology" dataset has the most number of attributes for the category, and "Post-operative" has the smallest number of attributes.

Table 4.2  
Binomial data distribution

Name	Class	
	0	1
Hepatitis	20.6%	79.4%
SPECT Heart	20.6%	79.4%
SPECTF Heart	20.6%	79.4%
Haberman's Survival	73.5%	26.5%
Parkinson's Survival	75.4%	24.6%
Thoracic Surgery	14.9%	85.1%

Table 4.2 shows the distribution of binomial datasets. The datasets are ordered according to their imbalance percentage. The imbalance pairings for the datasets are 15-85, 20-80, 20-80, 74-26 and 75-25. The distribution ratios are not balanced either by 50-50 or 40-60 pairing. The datasets with huge gap of ratio is more exposed to encounter biased prediction as compared to a dataset with smaller ratio gap. The class

with the most instances is called "majority class" and the class with the least instances is called "minority class". The datasets have two classes for the instances. For binomial datasets, the majority class and minority class is apparent.

Table 4.3  
Multiclass data distribution

Name	Majority Class	Minority Class	Other Classes	Total Class
Primary Tumor	24.8%	0%	75.2%	22
Zoo	40.6%	4.0%	55.4%	7
Ecoli Protein Localization	42.5%	0.4%	57.1%	8
Yeast Protein Localization	31.2%	0.3%	68.5%	10
Standardized Audiology	24%	0.5%	75.5%	24
Contraceptive Method Choice	42.7%	22.6%	34.7%	3
Dermatology	30.6%	5.5%	63.9%	6
Post-operative	71.1%	2.2%	26.6%	3

Table 4.3 presents the data distribution for multiclass datasets, which means the instances belong to one of many classes. The imbalance ratio can be seen in the data distribution for each class. The dataset with the most classes is the "Standardized Audiology" dataset. It has 24 classes. The datasets with the least classes are the "Contraceptive Method Choice" and "Post-operative" datasets. Both datasets have 3 classes.

The data are divided into nominal and ordinal categories. Nominal is when a dataset label has more than two classes. The instances are sorted into each classes according to the attributes it satisfies. For example, "Ecoli Protein Localization" dataset in nominal category has 8 classes. The classes are labeled into "cp" for cytoplasm, "im" for inner membrane without signal sequence, "pp" for periplasm, "imU" for inner membrane with uncleavable signal sequence, "om" for outer membrane, "omL" for outer membrane lipoprotein, "imL" for inner membrane lipoprotein and "imS" for inner membrane with cleavable signal sequence. The attributes of the imbalanced class dataset is used as input for an algorithm to learn and extract the features in order to predict the output according to the class it belongs to.

In ordinal category, the instances are also sorted into each classes according to

the attributes it satisfies. For example, "Contraceptive Method Choice" dataset has 3 classes, which is labeled "1", "2" or "3". Label "1" is when the subject is not using a contraception, label "2" is when the subject is using a contraception in long-term period and label "3" is when the subject is using a contraception in short-term period. However, the classes are linked to each other in varying degrees in ordered form. On top of that, the attributes are labeled from "Low" to "High" as compared to datasets in nominal category which labels the attributes in numerical or label without explicit arrangement. Another example is the "Dermatology" dataset, where the instances are classified into 6 classes labelling the skin condition that has very little different stage of severity.

In Table 4.3, the data distribution for multiclass category is not presented according to the classes, unlike binomial category in Table 4.2. Instead, it presents the distribution of the majority class, minority class and other classes in the dataset. This is because all the imbalanced class datasets in multiclass category has varied number of classes. Therefore, it might not clear to see the difference between the majority and minority classes if the data distribution is presented according to each classes. From Table 4.3, the data disparity difference can be seen clearly.

### **4.3 METRICS EVALUATION**

The performance of Evolutionary DBN with bootstrap sampling is measured based on its comparison to other algorithms. The algorithms used for comparison are:

1. Deep belief network (DBN)
2. Deep neural network (DNN)
3. Backpropagation neural network (BPNN)

#### 4. Support vector machine (SVM)

The performance metrics used to assess the performance of the algorithm are:

1. Confusion Matrix
2. Accuracy rate
3. Weighted mean recall
4. Weighted mean precision
5. F1-score
6. AUC

The description of each performance metric is detailed in Section 4.3.1. Statistical tests are also implemented as evaluation for Evolutionary DBN with bootstrap sampling's performance when handling imbalanced class datasets. The statistical tests used for the experiment in this thesis are Wilcoxon Signed Rank test and Vargha Delaney A test. The details of the statistical test is described in Section 4.3.2

##### **4.3.1 Performance Metrics**

The details of the performance metrics used to evaluate and compare the performance of evolutionary DBN to other deep learning and machine learning algorithms are explained as below. Each metric has its merits and limitations when measuring an algorithm model. Hence, the overall performance are taken into account and analyzed instead of just using a single metric as an evaluation approach.

#### 4.3.1.1 Confusion Matrix

When a model is used to predict a dataset, the data is split into training and testing. Training data is utilized for the model to train and testing data is used to measure whether the model manage to predict accurately. The outcome of a classification task is categorized into several categories which are (H. Han et al., 2005):

1. True positive (TP)
2. True negative (TN)
3. False positive (FP)
4. False negative (FN)

		PREDICTED	
ACTUAL			
		TP	FP
		FN	TN

Figure 4.2: Representation of a confusion matrix

Based on Figure 4.2 above, the type of categories is determined according to the predict and actual relation matrix. For example, when an algorithm predicted the value of '0', and the actual value is '0', then it is categorized into TP. If the actual value is '1' instead, it is categorized into FP. When an algorithm predicted the value '1', and the actual value is '1', then it is categorized into TN, whereas if the actual value is '0', it is categorized into FN. This is the evaluation for binary class datasets.

For multiclass datasets, the TP is the correct class predicted for all class. The FN of a class is the sum of values in the corresponding row of the matrix excluding

TP. The number of FP for a class is the sum of values of in the corresponding column excluding TP. The number of TN for a certain class is the sum of all the columns and rows excluding the particular class' column and row in the matrix.

These outcomes are the basis of the computation for other metrics used in this thesis. It is a raw representation to the number of correctly predicted in each class. It is a good measure to see which class is mostly used for prediction and which class is often correctly or wrongly predicted.

#### **4.3.1.2 Accuracy Rate**

The calculation of accuracy rate is straightforward. It is the number of correct prediction out of all the testing set. The formula is depict as follows;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

The higher the accuracy value, the better the algorithm performs. However, in the case of imbalanced class dataset classification, accuracy rate alone is not enough to determine the achievement of an algorithm. This is because there is a possibility of a scenario where the instances are mostly made up of majority class instead of minority class. This can pose a problem when the minority class is needed more than the majority class.

A common example for this case is detecting credit card fraud, where the minority class consists of the events of actual credit card fraud as compared to the majority class. Hence, other performance metrics are taken into consideration to assess the performance of Evolutionary DBN with bootstrap sampling as a whole.

#### 4.3.1.3 *Weighted Mean Recall*

Recall is a metric evaluation that assesses the rate of recall of an algorithm. When an algorithm is testing the input, it has to ensure a high level of recall. The recall score range from [0,1]. This shows that the algorithm is robust and not biased. In imbalanced class issue, it is crucial for an algorithm to have a good recall score because that indicates the algorithm manage to 'recall' many, if not all, instances as training and testing (Qiong et al., 2009). The algorithm is considered to be unbiased.

The formula for recall is presented in Formula 4.2. The formula of weighted mean recall is presented in Formula 4.3

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

$$WMR = \frac{1}{\sum_{l \in L} |\hat{y}_l|} \sum_{l \in L} |\hat{y}_l| \phi(y_l, \hat{y}_l) \quad (4.3)$$

#### 4.3.1.4 *Weighted Mean Precision*

Precision evaluates an algorithm's preciseness and exactness. If an algorithm manage to recall many instances, but is not precise, then the algorithm is not performing well. An algorithm must score a high level of precision to determine its performance. The formula of a weighted mean precision is similar to Formula 4.3, but is calculated using Formula 4.4

$$Precision = \frac{TP}{TP + FP} \quad (4.4)$$

#### 4.3.1.5 F1-Score

F1-score determines the harmonic value between recall and precision values of an algorithm. For example, an algorithm might have a high score of precision but low score of recall would mean the model is biased. In imbalanced class problem, it could mean the algorithm is only recalling the instances from majority class. However, if the algorithm has a high score of recall but low score of precision would indicate that the algorithm is not accurate enough despite manage to use all the instances for testing.

F1-score levels the contrast between recall and precision and see if the results tally to each other. The higher the F1-score, indicates that the recall and precision agrees with each other and the algorithm is performing well.

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.5)$$

#### 4.3.1.6 Area Under Curve

To calculate area under curve (AUC), the value of true positive rate (TPR) and false positive rate (FPR) must be determined first. The formulas for TPR and FPR are shown in Formula 4.6 and Formula 4.7 respectively. Then, a graph is plotted as shown in Figure 4.3. AUC ranges from [0,1]. If the area reaches 1, it indicates the algorithm is performing well.

$$TPR = \frac{TP}{FN + TP} \quad (4.6)$$

$$FPR = \frac{FP}{FP + TN} \quad (4.7)$$



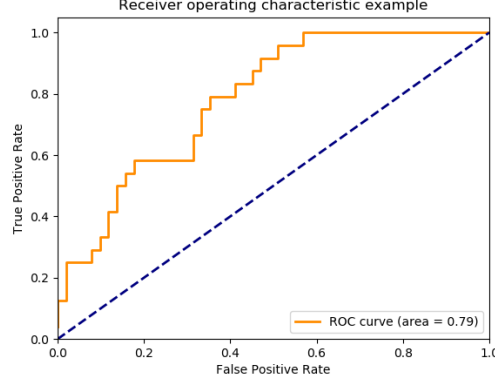


Figure 4.3: Example of AUC plot.

As shown in Figure 4.3, TPR is plotted against FPR. AUC is a competent metric, especially for imbalanced class datasets (Davis & Goadrich, 2006). This is because AUC is inclusive with the data distribution, therefore the data disparity between majority and minority classes are taken into account for the calculation.

#### 4.3.2 Statistical Tests

A statistical test is a quality method to determine the algorithm to perform well due to its own attribute and not by coincidence. There is two kinds of statistical tests, which are parametric test and non-parametric test. A parametric test makes strong assumptions, while non-parametric makes weaker assumption as it has to validate all assumptions. A statistical test is selected based on the problem type and classifier information at hand.

##### 4.3.2.1 Wilcoxon Signed Rank Test

Wilcoxon Signed Rank Test is a non-parametric test. It tests two classifiers with multiple domains. The experiment in this thesis includes the comparison between the actual output and predicted output of each imbalanced class datasets handled by Evolutionary DBN with bootstrap sampling. In addition, the imbalanced class datasets utilized for

the experiment consists of multiple domains. Demsar (2006) recommends Wilcoxon Sign Rank Test as a mean to compare different algorithms with different datasets. This is a good assessment method for the proposed algorithm's performance.

The difference of performance between the actual output and predicted output compared are calculated. Then, the absolute values of the differences are ranked in increasing value. Next, the sum of positive and negative ranks are calculated and contrasted using the formula,

$$T_{wilcox} = \min(W_{S1}, W_{S2}) \quad (4.8)$$

where  $W_{S1}$  is the positive rank and  $W_{S2}$  is the negative rank. The minimum value between the two rank values are chosen and compared to critical value,  $V_\alpha = 0.05$ . If the value of  $T_{wilcox}$  is less than or equal to the critical value, the null hypothesis that states the actual output and predicted output have the same performance is rejected. Otherwise, it is accepted.

#### 4.3.2.2 Vargha-Delaney A Test

Vargha-Delaney A Test is another non-parametric test used to evaluate the performance of Evolutionary DBN with bootstrap sampling (Vargha & Delaney, 2000). The comparison of actual output and predicted output of the algorithm is taken and compared whether there is a significant difference between the two result. The A test computes the rank sum and then computed a measure based on the rank sum result.

The A-measure is computed using the formula,

$$A = \frac{\left(\frac{r_1}{m} - \frac{(m+1)}{2}\right)}{n} \quad (4.9)$$

where  $r_1$  is the value of rank sum between actual output and predicted output,  $m$

is the length of actual output and  $n$  is the length of predicted output. The evaluation of the measure value is as presented as below:

1. Around 0.5 = No difference
2. Around 0.56 = Small difference
3. Around 0.64 = Medium difference
4. Around 0.71 = Large difference

This test is used as performance evaluation for Evolutionary DBN with bootstrap sampling because of its approach of comparing the differences of the actual and predicted outputs. This approach is more sensitive to rate the raw differences.

## **CHAPTER 5**

### **EVOLUTIONARY DEEP BELIEF NETWORK WITH BOOTSTRAP SAMPLING**

#### **5.1 INTRODUCTION**

This section explains the proposed optimized algorithm used in the thesis. The optimized deep learning algorithm implemented in this thesis is an Evolutionary deep belief network (DBN) with bootstrap sampling. The original DBN algorithm is explained in Section 5.2 as a comparison for the modifications performed on the proposed algorithm. The details and modification aspect of the proposed algorithm is described in Section 5.3

#### **5.2 DEEP BELIEF NETWORK**

Algorithm 5.1 presents the pseudocode of DBN. Algorithms 5.2 to 5.5 explains the pretraining and finetuning aspects of DBN in detailed manner. Figure 5.1 presents the flowchart of DBN algorithm for summarization. RBM is computed for pretrain phase while BPNN is utilized for finetuning phase.

---

**Algorithm 5.1** Deep Belief Network

---

**Input** Dataset**Output** PerformanceMetrics

```
1: dataset  $\leftarrow$  train_set, test_set
2: train_set  $\leftarrow$  0.8
3: test_set  $\leftarrow$  0.2
4: maxEpoch  $\leftarrow$  100
5: classifier  $\leftarrow$  DBN
6: for DBNclassifier do
7:   hidden_layer  $\leftarrow$  [256, 256]
8:   n_epoch_rbm  $\leftarrow$  5
9:   activation_function  $\leftarrow$  'relu'
10:  n_iteration_bp  $\leftarrow$  100
11:  learning_rate  $\leftarrow$  0.1
12:  batch_size  $\leftarrow$  32
13:  dropout_p  $\leftarrow$  0.2
14: end for
15:  classifier_train  $\leftarrow$  train_set
16:  classifier_predict  $\leftarrow$  test_set
```

---

In Algorithm 5.1, the model receives an imbalanced class dataset as input. The instances of the data is taken and split up into training and testing set using a cross validation. The testing set of the dataset is assigned to 0.2, which means the 80% of the dataset is used for training and 20% is left out and is used for testing. The maximum epoch is set to 100. One epoch will take the inputs as neurons and calculate the weights and biases into the connected hidden layers. One node of a neuron consists of initialized weights and are calculated with activation function. The output of one neuron is an input for another neuron, which allows the network to learn.

As explained in literature review done in Section 2.5.1, DBN is made up of stacked Restricted Boltzmann Machine (RBM). The input will go through a network of connected RBM for pre-training phase and backpropagation neural network (BPNN) for fine-tuning phase. Then, the weights will be adjusted for the next epoch as per trained by the previous epoch. After the calculated weights are adjusted, the network will iterate the same process of calculation and keep adjusting the weights until the maximum epoch

is reached. The weights are responsible for the network to make decision and predict the output as trained using the training data.

In the algorithm, pre-training using RBM is set to 5. The DBN classifier is assigned with 2 hidden layers that consists of 256 neurons per hidden layer. Amri et al. (2017) performed DBN analysis and explains that the number of hidden layer only effect computation time, but not significant on accuracy rate increment. An activation function is important for DBN to be used by the neurons in the hidden layer . The activation function used for this algorithm is the rectifier linear unit (ReLU). ReLU is a common activation function used for deep learning algorithms. The formula for ReLU is

$$f(x) = \max(0, x) \quad (5.1)$$

Other types of activation functions are sigmoid and hyperbolic tangent (tanh). Sigmoid function has the formula

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.2)$$

It has the range of (0,1). The disadvantage of sigmoid activation function is it can prompt the network to get stuck during training. Tanh has the formula

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (5.3)$$

The range of tanh is (-1,1), which provides accurate representation between negative and positive values. ReLU is chosen as the activation function for evolutionary DBN because it provides better performance metrics as compared when sigmoid and tanh is used (Amri et al., 2017; Dahl et al., 2013; X. Zhang et al., 2014). M. Chen et al. (2014)

recited the ability of ReLU to converge quicker for deep learning models as compared to its sigmoid counterpart when handling the same problem.

Backpropagation for finetuning epoch is set to 100. The learning rate of the algorithm is set to 0.1. Meanwhile, the batch size and dropout number is set to 32 and 0.2 respectively. The algorithm performs the classification task on the imbalanced class datasets based on these set parameters and produce the performance metrics as explained in Section 4.3.1. The details of components in DBN is shown in Algorithms 5.2 to 5.5

---

**Algorithm 5.2** BinaryRBM

---

**Input** Data from Deep Belief Network

**Output** BinaryRBM

```

1: hidden_units  $\leftarrow$  100
2: activation_function  $\leftarrow$  'sigmoid'
3: optimization  $\leftarrow$  'sgd'
4: for BinaryRBM do
5:   initialization_RBM  $\leftarrow$  hidden_units, activation_function, optimization
6:   visible_layers  $\leftarrow$  initialization_RBM
7:   hidden_layers  $\leftarrow$  initialization_RBM
8:   fit_RBM  $\leftarrow$  hidden_layers, visible_layers, Data
9:   transform_RBM  $\leftarrow$  fit_RBM
10:  reconstruct_RBM  $\leftarrow$  transform_RBM
11:  ReconstructErrorComputation  $\leftarrow$  transform_RBM, reconstruct_RBM
12: end for

```

---

Algorithm 5.2 shows the functions and structure of RBM modules used to make up the layers for pretrain phase in DBN. The RBM modules are initialized . Stochastic gradient descent (SGD) is used as optimization algorithm, whereas contrastive divergence method is employed for the gradient computation. The visible and hidden layers are constructed based on the hidden units assigned. The model is then fit based on the data received as input. The fitted model is then transformed and later is used for reconstruction of the modules. Reconstruction error computation is also performed to ensure the adjustment of the weights for RBM visible and hidden layers. This structure is then

used as part of pretraining in DBN.

---

**Algorithm 5.3** UnsupervisedDBN

---

**Input** BinaryRBM

**Output** UnsupervisedDBN

```
1: hidden_layers  $\leftarrow$  [100, 100]
2: activation_function  $\leftarrow$  'sigmoid'
3: optimization  $\leftarrow$  'sgd'
4: rbm_class  $\leftarrow$  BinaryRBM
5: for UnsupervisedDBN do
6:   initialization_unsDBN  $\leftarrow$  hidden_layers, activation_function,
     optimization, rbm_class
7:   fit_unsDBN  $\leftarrow$  initialization_unsDBN, Data
8:   transform_unsDBN  $\leftarrow$  fit_unsDBN
9: end for
```

---

Algorithm 5.3 presents the pretraining process of DBN. The initialization phase set the hidden layers into two with 100 units each. Activation function used is sigmoid function. SGD is used as optimization algorithm and the RBM modules computation used is the one initialized in Algorithm 5.2. After the initialization phase of DBN, the model is then fitted using the dataset and then the data is transformed based on the hidden units computation in Algorithm 5.2. The whole process is used as pretraining phase in DBN as shown in Algorithm 5.4

---

**Algorithm 5.4** AbstractSupervisedDBN

---

**Input** UnsupervisedDBN, *fit\_unsDBN*

**Output** NumPyAbstractSupervisedDBN

```
1: hidden_layers  $\leftarrow$  [100, 100]
2: activation_function  $\leftarrow$  'sigmoid'
3: optimization  $\leftarrow$  'sgd'
4: batch_size  $\leftarrow$  32
5: pretrain  $\leftarrow$  fit_unsDBN
6: initialization  $\leftarrow$  hidden_layers, activation_function,
     optimization, batch_size, UnsupervisedDBN
7: fit_supDBN  $\leftarrow$  pretrain, Data
8: predict_supDBN  $\leftarrow$  fit_supDBN
9: transform_unsDBN  $\leftarrow$  fit_supDBN
```

---

Algorithm 5.4 presents the phase where DBN used the pretraining computations



performed in Algorithm 5.3. The supervised DBN initialized its 2 hidden layers with 100 neurons each, with sigmoid activation function and used SGD as optimization algorithm. The batch size is set to 32. The initialization in unsupervised DBN in Algorithm 5.3 is utilized as a part of initialization in supervised DBN. The algorithm then fit the dataset and perform prediction. Then it will transform the fitted model of supervised DBN part which includes the pretraining of unsupervised DBN.

---

**Algorithm 5.5** NumPyAbstractSupervisedDBN

---

**Input** AbstractSupervisedDBN, Data

**Output** DBNclassifier

- 1:  $input\_vector, label \leftarrow Data$
  - 2:  $SGD \leftarrow Data$
  - 3:  $compute\_activation \leftarrow Data$
  - 4:  $backpropagation \leftarrow input\_vector, label$
  - 5:  $finetuning \leftarrow backpropagation$
- 

Algorithm 5.5 presents the pseudocode of finetuning phase in DBN. The input vector and label is assigned from the dataset. SGD is performed on the dataset input as optimization algorithm as well. The training loss of the network is provided based on SGD computation. Activation computations requires the input data to compute output values of all layers. BPNN is computed based on the input vector and labels from the input data and is used as the finetuning for classification task.

The summary of process of DBN is represented in flowchart form in Figure 5.1 below. The calculation of SGD and BPNN act as the finetuning for the pretrained unsupervised and supervised DBN made up of the initialization of binary RBM. The setting of the processes occur in DBN algorithm is obtained from the DBN classifier in the main program.

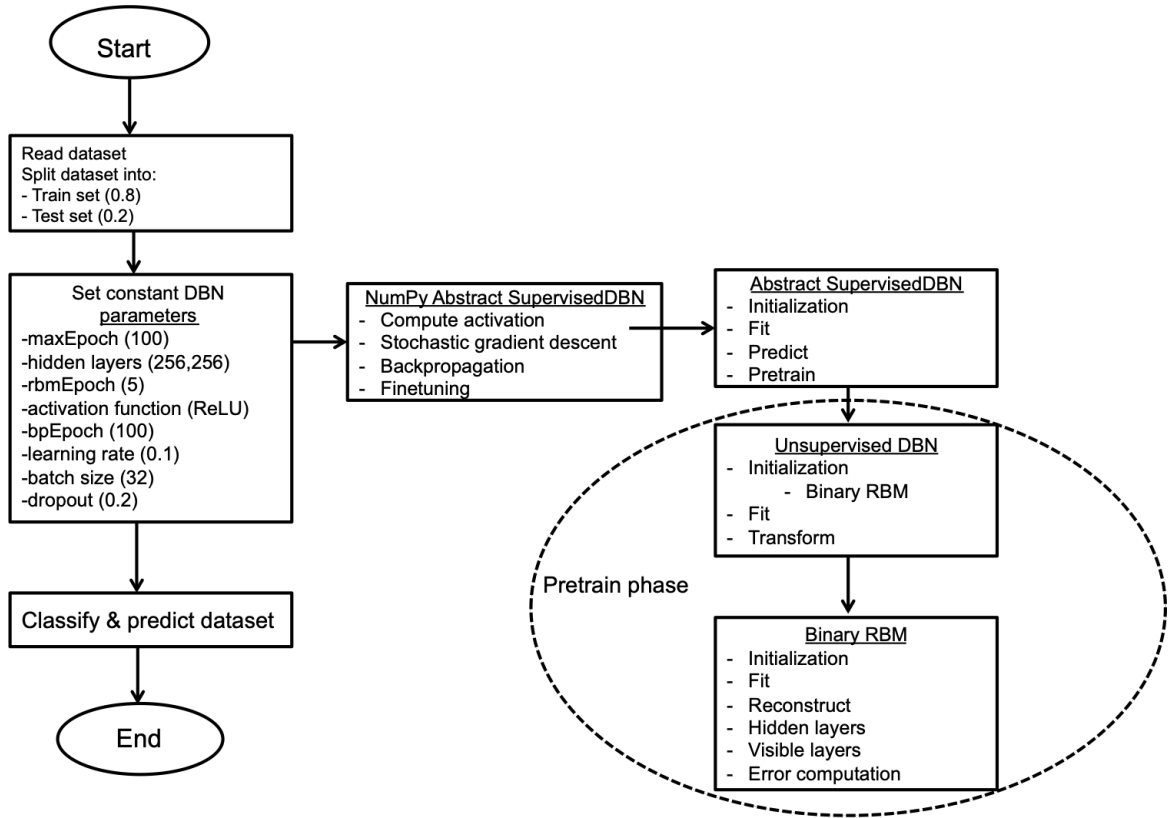


Figure 5.1: Flowchart of DBN

### 5.3 EVOLUTIONARY DEEP BELIEF NETWORK WITH BOOTSTRAP SAMPLING

Algorithm 5.6 presents the pseudocode of Evolutionary DBN with bootstrap sampling. As discussed in Algorithm 5.1, the flow of Evolutionary DBN with bootstrap sampling is the same as the original DBN process with difference of incorporation of evolutionary algorithm and bootstrap sampling. In Evolutionary DBN with bootstrap sampling, a few parameters used for DBN classifier is determined by genetic algorithm (GA). After the whole network training, the optimized algorithm goes through bootstrap sampling. The performance metrics are obtained after the optimized algorithm finishes the bootstrap sampling phase.

Further explanation for GA utilized as evolutionary algorithm as optimization

feature for Evolutionary DBN with bootstrap sampling is shown in Algorithm 5.7. Algorithms 5.8, 5.9, 5.10 and 5.11 describes the sections of GA in detailed manner.

---

**Algorithm 5.6** Evolutionary Deep Belief Network with Bootstrap Sampling

---

**Input** Dataset

**Output** PerformanceMetrics

```

1: dataset  $\leftarrow$  train_set, test_set
2: train_set  $\leftarrow$  0.8
3: test_set  $\leftarrow$  0.2
4: maxEpoch  $\leftarrow$  100
5: classifier  $\leftarrow$  DBN
6: for DBNclassifier do
7:   hidden_layer  $\leftarrow$  [256, 256]
8:   n_epoch_rbm  $\leftarrow$  5
9:   activation_function  $\leftarrow$  'relu'
10:  n_iteration_bp  $\leftarrow$  GA
11:  learning_rate  $\leftarrow$  GA
12:  batch_size  $\leftarrow$  GA
13:  dropout_p  $\leftarrow$  GA
14: end for
15:  classifier_train  $\leftarrow$  train_set
16:  random_shuffle  $\leftarrow$  dataset
17:  classifier_train  $\leftarrow$  train_set
18:  random_shuffle  $\leftarrow$  dataset
19:  classifier_train  $\leftarrow$  train_set
20:  random_shuffle  $\leftarrow$  dataset
21:  classifier_train  $\leftarrow$  train_set
22:  classifier_predict  $\leftarrow$  test_set

```

---

Genetic algorithm (GA) is employed as the evolutionary part of Evolutionary DBN with bootstrap sampling. The main steps of GA are initializing population, calculate the fitness of the individuals in current population, creating new population and the fitness of its individuals are calculated and compared with previous population. Next, the individuals with best fitness value will be mutated and evolved for the next generation. The procedure is repeated until termination. Details of GA is explained with the representation of Algorithm 5.7.

Bootstrap sampling is implemented when an evolutionary DBN classifier has optimized its parameter setting. The classifier will train itself using the initial training

data split. Then, the dataset gets reshuffled using the same ratio utilized for the initial training data. The classifier will retrain itself again. This process repeats until the fourth time. This is the optimal sampling number for Evolutionary DBN. As the maximum epoch reached, the testing set is used to test the algorithm performance. The algorithm predicts and classify the output according to its learning performance. Performance metrics such as confusion matrix, accuracy rate, weighted mean precision, weighted mean recall, F1-score and AUC is computed and taken as measure to evaluate the overall implementation of Evolutionary DBN with bootstrap sampling.

---

**Algorithm 5.7** Genetic Algorithm

---

**Input**  $n\_iteration\_bp$ ,  $learning\_rate$ ,  $batch\_size$ ,  $dropout\_p$

**Output**  $Final\_population$

```

1:  $size \leftarrow 5$ 
2:  $First\_population \leftarrow create\_population$ 
3: for  $create\_population$  do
4:   while  $i < size$  do
Require:  $DBNclassifier$ 
5:      $hidden\_layer \leftarrow [256, 256]$ 
6:      $n\_epoch\_rbm \leftarrow 5$ 
7:      $activation\_function \leftarrow 'relu'$ 
8:      $n\_iteration\_bp \leftarrow random(100, 300)$ 
9:      $learning\_rate \leftarrow random(0.01, 0.1)$ 
10:     $batch\_size \leftarrow random(1, 10)$ 
11:     $dropout\_p \leftarrow random(0.1, 0.6)$ 
12:  end while
13: return  $First\_population$ 
14: end for
15:  $Evolve\_population \leftarrow First\_population$ 
16:  $Final\_population \leftarrow Evolve\_population$ 
return  $Final\_population$ 

```

---

In Evolutionary DBN with bootstrap sampling, GA works as follows. Firstly, the algorithm will create an initial population consists of DBN classifier with parameters that are assigned randomly. The parameters are the BPNN iteration number for fine-tuning, the learning rate, batch size and the dropout of the algorithm. BPNN is a conventional machine learning algorithm that has stable learning method. However,

BPNN is commonly known to get stuck at local minima after some time(Lanbouri & Achchab, 2015). Therefore, the number of iteration is set randomly between 100 to 300.

The learning rate is responsible for the weight adjustments of the network. This role will avoid the algorithm to hit plateau. If the learning rate is too low, the learning will increase computational time, whereas, if the learning rate is set too high, it has the probability of missing the optimal mark. Hence, the learning rate is assigned randomly between 0.01 and 0.1 as adopted by Jo & Lee (2015) when setting the ideal learning rate.

Batch size is the number of training examples utilized in one iteration. It is set between 1 to 10 in order to avoid bias in reusing the same training examples. Dropout is a regularization technique to prevent overfitting in the algorithm. It is set randomly between 0.1 and 0.6. The population size is set to 5, therefore, there are 5 DBN classifiers with fixed and randomized parameters as explained previously.

This initial population will be used as a part of evolution process, and the process will return the final population to the Evolutionary DBN with bootstrap sampling algorithm.

---

**Algorithm 5.8** Evolve Population

---

**Input** First\_population

**Output** evolved\_population

- 1: *Calculate\_Fitness*  $\leftarrow$  *First\_population*
  - 2: *create\_new\_generation*  $\leftarrow$  *calculated\_fitness\_population*
  - 3: *mutation*  $\leftarrow$  *new\_generation*
- 

Algorithm 5.8 explains the procedure of the evolution process in GA. This part will calculate the fitness of the population.

The population member with the best fitness value is chosen and is a part of a new generation. Next, this new generation consisted of a population of members with

good fitness values are mutated. This new generation acts as the final population in Algorithm 5.7.

---

**Algorithm 5.9** Calculate fitness

---

**Input** First\_population

**Output** sorted\_population

```

1: for  $i$  do population-1
2:    $train \leftarrow i$ 
3:    $test \leftarrow i$ 
4:    $predict \leftarrow i$ 
5: return  $accuracy_i$ 
6: end for
7:  $sort\_by\_accuracy$ 

```

---

Algorithm 5.9 explains the fitness calculation of GA. For the individual in the population, it will go through training and testing phase. The the accuracy score of the individual is recorded and compared with the accuracy score of other individuals of the same population. After each individual gone through the process, they will be sorted from the highest accuracy to the lowest accuracy. The fitness function is shown in Equation 5.4

$$f(x) = sort(x_1, x_2, \dots, x_i) \quad (5.4)$$

where,

$x$  is the accuracy rate,

$i$  is the population size

After the fitness evaluation, the chromosome selection is occurred. The new generation in Algorithm 5.10 is created using the population with sorted fitness value. The fittest individual is set to variable "father" and the second fittest individual is set to variable "mother". These two individuals will be the first to populate the new generation

population for crossover purposes. The difference between mutation and crossover is crossover involves a recombination of two individuals, whereas mutation is applied to a single individual in the population. The type of crossover performed is the multipoint crossover.

---

**Algorithm 5.10** Create new generation

---

**Input** sorted\_population

**Output** new\_generation

```

1:  $father \leftarrow sorted\_population[0]$ 
2:  $mother \leftarrow sorted\_population[1]$ 
3:  $next\_generation[] \leftarrow father, mother$ 
4:  $number\_of\_children \leftarrow size - len(next\_generation)$ 
5: for child do number_of_children
Require: DBNclassifier
6:    $hidden\_layer \leftarrow [256, 256]$ 
7:    $n\_epoch\_rbm \leftarrow 5$ 
8:    $activation\_function \leftarrow 'relu'$ 
9:    $n\_iteration\_bp \leftarrow random(father_{n\_iteration\_bp}, mother_{n\_iteration\_bp})$ 
10:   $learning\_rate \leftarrow random(father_{learning\_rate}, mother_{learning\_rate})$ 
11:   $batch\_size \leftarrow random(father_{batch\_size}, mother_{batch\_size})$ 
12:   $dropout\_p \leftarrow random(father_{dropout\_p}, mother_{dropout\_p})$ 
13: end for
return new_generation

```

---

Multipoint crossover occurs when the child chromosomes copied from one parent at one crossover point and another parent at next crossover point. Presented in Algorithm 5.10, the crossover points would be the BPNN iteration number, the learning rate, the batch size and the dropout number. The child chromosome acquires the variable parameter from either the "father" or "mother". This phase returns the new generation required for mutation.

---

**Algorithm 5.11** Mutation

---

**Input** new\_generation**Output** mutated\_population

```
1: variables  $\leftarrow$  n_iteration_bp, learning_rate, batch_size, dropout_p
2: for individual do next_generation
3:   if variable  $\leftarrow$  dropout_p then
4:     setdropout_p  $\leftarrow$  random(0.1, 0.6)
5:   else
6:     if variable  $\leftarrow$  n_iteration_bp then
7:       setn_iteration_bp  $\leftarrow$  random(100, 300)
8:     else
9:       if variable  $\leftarrow$  learning_rate then
10:        setlearning_rate  $\leftarrow$  random(0.01, 0.1)
11:      else
12:        if variable  $\leftarrow$  setbatch_size then
13:          setbatch_size  $\leftarrow$  random(1, 10)
14:        end if
15:      end if
16:    end if
17:  return mutated_population
```

---

Algorithm 5.11 presents the mutation process of GA. The individuals in the new generation created in Algorithm 5.10 scrutinize its variables to be optimized. In this scenario, the variables are the number of backpropagation iteration, learning rate, batch size and dropout size. The variables are chosen randomly and each the variables are set to random within its range.

After every individual has their mutated variables, they will be returned to Algorithm 5.8 as mutated population. Then, Algorithm 5.8 performs an evolution process as explained previously and return the evolved population to GA in Algorithm 5.7.

The summary of the modification done for Evolutionary DBN with bootstrap sampling is represented in Figure 5.2.



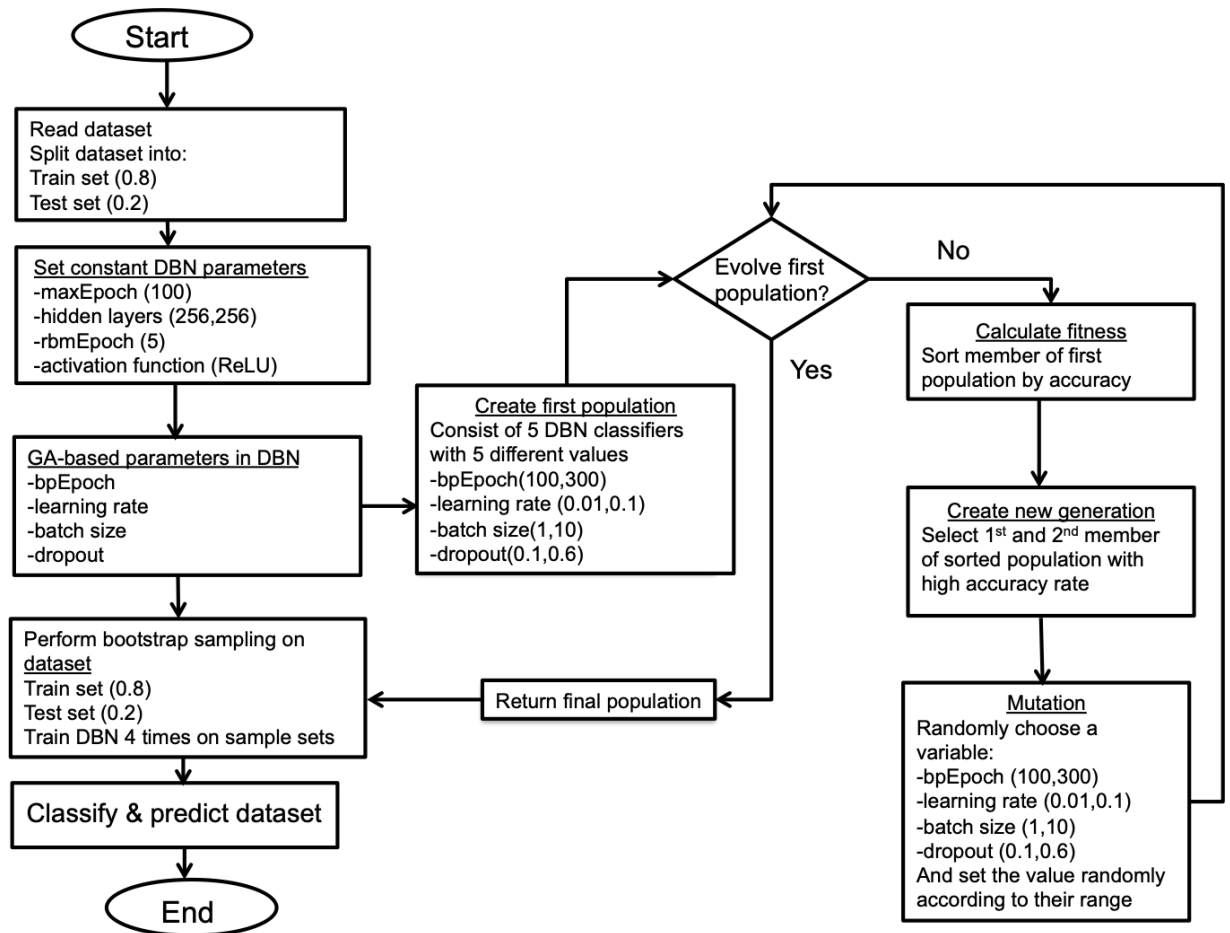


Figure 5.2: Flowchart of Evolutionary DBN with bootstrap sampling

## **CHAPTER 6**

### **RESULTS AND DISCUSSION**

#### **6.1 INTRODUCTION**

This chapter outlines the results and analysis of the experiment conducted in Chapter 4. This chapter is breakdown into two part. The first part describes the performance metrics of Evolutionary DBN with bootstrap sampling and its comparison to deep learning algorithms such as DBN and DNN. The comparison between Evolutionary DBN with bootstrap sampling and machine learning algorithms such as BPNN and SVM is also presented. The results are depicted in tabular form and then analyzed. Evolutionary DBN with bootstrap sampling is referred to "Evo-sDBN" in the tables. The second part presents the statistical tests done on Evolutionary DBN with bootstrap sampling and its analysis when managing imbalanced class datasets.

#### **6.2 PERFORMANCE METRICS RESULTS**

This section presents the performance metrics of Evolutionary DBN with bootstrap sampling and is compared to deep learning and machine learning algorithms. As mentioned in Section 4.3.1, the performance metrics used to measure and compare the performance of the algorithms are:

1. Confusion matrix
2. Accuracy rate
3. Weighted mean recall
4. Weighted mean precision

## 5. F1-score

## 6. Area under curve

The results are presented in tabular form and are analyzed. Evolutionary DBN with bootstrap sampling outperforms the deep learning and machine learning algorithms used as comparison. The nature of imbalanced class dataset is also taken into account as a factor that influenced the evolutionary DBN with bootstrap sampling performance.

### 6.2.1 Confusion Matrix

Table 6.1 presents the result of the confusion matrix obtained by the deep learning and machine learning algorithms. As mentioned in Section 4.2, the imbalanced class datasets are categorized into binomial, nominal and ordinal. For each of the category, the datasets are labeled according to its attribute characteristics such as binary, discrete numerical, continuous numerical and multivariate.

In binomial category, the confusion matrix is based on the class distribution described in Table 4.2. For "Hepatitis" dataset, Evolutionary DBN with bootstrap sampling is leaning to have high number of correctly predicted in TP category, which happens to be the minority class. DBN, BPNN and SVM leaned towards TN. For "SPECT Heart" dataset, Evolutionary DBN with bootstrap sampling has high number of TN, which belongs to the majority class. The same is achieved by DNN, BPNN and SVM. However, DBN has small number difference between TP and TN.

For "SPECTF Heart" dataset, Evolutionary DBN with bootstrap sampling is leaning towards TN as the "SPECT Heart" dataset. For "Haberman's Survival" dataset, Evolutionary DBN with bootstrap sampling has a high TP score. In "Parkinson's Survival" dataset, evolutionary DBN with bootstrap sampling has higher TN value. Evolu-

Table 6.1  
Confusion Matrix results of Algorithms

Category	Attributes	Characteristic	Name	Category	Evo-sDBN	DBN	DNN	BPNN	SVM
Binomial	Binary	Binary	Hepatitis	TP	16	0	3	0	0
				TN	0	13	0	14	14
				FP	0	3	0	2	2
			SPECT Heart	FN	0	0	13	0	0
				TP	2	24	9	12	13
				TN	51	17	45	41	41
			SPECTF Heart	FP	1	8	0	1	0
				FN	0	5	0	0	0
				TP	0	0	0	10	0
			Discrete Numerical	TN	50	45	45	23	41
				FP	4	9	3	9	13
				FN	0	0	0	18	0
			Discrete Numerical	TP	59	37	37	35	44
				TN	0	0	0	6	1
				FP	0	0	0	10	1
			Continuous Numerical	FN	3	25	25	11	16
				TP	0	0	0	6	0
			Multivariate	TN	38	29	29	1	32
				FP	1	10	10	1	7
				FN	0	0	0	31	0
Nominal	Binary	Multivariate	Thoracic Surgery	TP	83	76	76	83	83
				TN	0	0	0	0	0
				FP	0	0	0	0	0
			Primary Tumor	FN	11	18	18	11	11
				TP	24	7	7	1	1
				TN	0	17	0	24	25
			Zoo	FP	2	2	19	1	0
				FN	0	0	0	0	0
				TP	15	7	7	7	7
			Ecoli Protein Localization	TN	6	14	0	14	13
				FP	0	0	14	0	1
				FN	0	0	0	0	0
			Yeast Protein Localization	TP	67	35	0	29	30
				TN	0	16	33	32	34
				FP	1	17	0	6	4
Ordinal	Multivariate	Discrete Numerical	Standardized Audiology	FN	0	0	35	1	0
				TP	0	10	0	26	22
				TN	204	242	247	235	228
			Contraceptive Method Choice	FP	0	5	0	18	25
				FN	93	40	50	18	22
				TP	0	7	0	4	4
			Dermatology	TN	19	15	17	20	19
				FP	0	2	0	0	1
				FN	5	0	7	0	0
			Post-operative	TP	280	137	137	94	97
				TN	0	0	0	103	127
				FP	15	158	158	65	41
			Multivariate	FN	0	0	0	33	30
				TP	28	17	17	28	28
				TN	0	0	0	44	43
				FP	44	55	55	0	1
				FN	0	0	0	0	0

tionary DBN with bootstrap sampling has high TP value for "Thoracic Surgery" dataset.

For multiclass categories, the confusion matrix is based on the class distribution described in Table 4.3. In nominal category, Evolutionary DBN with bootstrap sampling is leaning the highest towards TP for "Primary Tumor", "Zoo" and "Ecoli Protein Localization" datasets, and has high TN value for "Yeast Protein Localization" and "Standardized Audiology" datasets.

For ordinal category, Evolutionary DBN with bootstrap sampling leans toward TP for "Contraceptive Method Choice" dataset and towards TN for "Post-operative" dataset. However, for "Dermatology" dataset, Evolutionary DBN with bootstrap sampling has highest value for FP category.

Further analysis is performed using the measurements of accuracy rate, weighted mean recall, weighted mean precision, F1-score and AUC score.

### **6.2.2 Accuracy Rate**

As explained in Section 4.3.1.2, the calculation of accuracy rate is the total number of correct prediction out of the total number of data predicted. The higher the value of accuracy rate shows the good performance of an algorithm. However, accuracy rate alone is not enough when imbalanced class datasets are involved because the algorithm might correctly classified the majority class of the dataset, but incorrectly classified the minority class of the dataset. Nevertheless, accuracy rate is still an important aspect to evaluate whether an algorithm is performing well or not.

Table 6.2 depicts the accuracy rate achieved for each algorithm for each imbalanced class dataset used in this experiment. The result is discussed according to the category of the imbalanced class datasets.

For binomial category, the proposed Evolutionary DBN with bootstrap sampling

Table 6.2  
Accuracy Rate of Algorithms

Category	Attributes	Characteristic	Name	Evo-sDBN	DBN	DNN	BPNN	SVM
Binomial	Binary		Hepatitis	<b>100%</b>	81.25%	18.75%	87.50%	87.50%
	Binary		SPECT Heart	<b>100%</b>	83.33%	<b>100%</b>	98.15%	<b>100%</b>
	Discrete Numerical		SPECTF Heart	<b>96.30%</b>	83.33%	83.33%	61.11%	75.92%
	Discrete Numerical		Haberman's Survival	<b>95.20%</b>	59.68%	59.67%	66.13%	72.6%
	Continuous Numerical		Parkinson's Survival	<b>97.44%</b>	74.36%	74.35%	17.95%	82.05%
	Multivariate		Thoracic Surgery	<b>94.70%</b>	80.85%	80.85%	88.30%	88.30%
Nominal	Binary		Primary Tumor	<b>100%</b>	53.85%	26.92%	69.23%	76.92%
	Binary		Zoo	95.24%	<b>100%</b>	33.33%	95.24%	90.50%
	Continuous Numerical		Ecoli Protein Localization	<b>100%</b>	67.65%	13.23%	77.94%	76.47%
	Continuous Numerical		Yeast Protein Localization	46.13%	48.15%	29.29%	<b>57.91%</b>	40.40%
	Multivariate		Standardized Audiology	<b>87.5%</b>	75.00%	72.00%	33.33%	70.83%
Ordinal	Discrete Numerical		Contraceptive Method Choice	<b>98.00%</b>	47.46%	46.44%	57.30%	63.05%
	Discrete Numerical		Dermatology	68.05%	58.33%	23.61%	<b>94.44%</b>	91.66%
	Multivariate		Post-operative	<b>94.44%</b>	72.22%	38.88%	72.22%	77.77%

achieved the highest accuracy rate contrast to other algorithms used for comparison, with exception for "SPECT Heart" dataset where DNN and SVM achieved a perfect accuracy score. SVM manage to score high accuracy rate for the dataset mentioned because the attributes of the dataset is in binary form, which is suitable for SVM hyperplane approach. Besides that, BPNN has accuracy of 98.15%, which is near perfect score. DBN also performed well for this dataset. It can be the structure of its attributes of this particular dataset is very commendable for classification task.

If we compare DBN result to other algorithm such as DNN, BPNN and SVM, the algorithm only score the highest when dataset "SPECTF Heart" is used in the experiment despite having the same accuracy rate as DNN. Therefore, Evolutionary DBN with bootstrap sampling manage to improve the performance of DBN when predicting the output of imbalanced binomial dataset.

Referring to Table 4.1 under Section 4.2, DBN manage to achieve the highest accuracy using "SPECTF Heart" data because it has the most attributes. Therefore, it is only ideal for DBN to perform when the algorithm has many features to learn from. Evolutionary DBN with bootstrap sampling overcome this specific requirement for DBN to perform and achieve the highest accuracy rate in all imbalanced class dataset in binomial category.

Both nominal and ordinal categories are considered multiclass with the difference where in ordinal category, the classes are ordered according to sequence.

In nominal category, Evolutionary DBN with bootstrap sampling attained the highest accuracy rate for imbalanced datasets, "Primary Tumor", "Ecoli Protein Localization" and "Standardized Audiology", which accounts for 3 out of 5 nominal datasets. For binary attributes in nominal category, Evolutionary DBN with bootstrap sampling accomplished high accuracy rate for "Primary Tumor" dataset, but second highest for

"Zoo" dataset alongside with BPNN. DBN scored the highest accuracy rate for "Zoo" dataset. Referring to Table 4.1, both of the imbalanced datasets has the same number of attributes which is 18, but different number of instances, which are 339 and 101 respectively.

Referring to Table 4.3, the total class for "Primary Tumor" dataset is 22 as opposed to 7 for "Zoo" dataset. For continuous numerical attributes in nominal category, Evolutionary DBN with bootstrap sampling accomplished high accuracy rate for "Ecoli Protein Localization" dataset, but second lowest for "Yeast Protein Localization" dataset. "Ecoli Protein Localization" has 8 attributes contrast to "Yeast Protein Localization" with 9 attributes. The instances are 336 and 1484 respectively. "Ecoli Protein Localization" dataset has 8 classes as compared to 9 classes of "Yeast Protein Localization" dataset. The parallel comparison between the binary and continuous numerical attributes shows that the number of instances, attributes, and total class of the imbalanced datasets are not the factors influencing the performance achieved by Evolutionary DBN with bootstrap sampling.

However, for the classes in "Zoo" dataset, although there are 7 classes, each of the class comprised of different animals that shares a similar attribute with each other but not necessarily from the same species. For example, Class 2 consists of 20 bird-like species such as "chicken", "penguin" and "vulture" among others. The shared attributes for these animals might be "2 legs" and "eggs", but a "penguin" is labeled "aquatic" as opposed to a "vulture" is labeled "airborne", while "chicken" is not labeled with such attributes. Yet, they are all considered in Class 2. This sort of structure in the dataset might cause Evolutionary DBN with bootstrap sampling unable to achieve 100% accuracy rate as compared to DBN.

For continuous numerical attribute, it is important to note that for "Yeast Protein



Localization" dataset, where Evolutionary DBN with bootstrap sampling achieved second lowest accuracy result yields very low accuracy rate from other algorithms as well with BPNN scored the highest at 57.91%.

In ordinal category, Evolutionary DBN with bootstrap sampling obtained the highest accuracy rate for 2 out of 3 imbalanced class datasets, which are "Contraceptive Choice Method" and "Post-operative". For "Dermatology" dataset, the algorithm achieved the second lowest accuracy result and DBN is the lowest accurate as compared to DNN, BPNN and SVM, which has between 91% and 95% accuracy range. Even though there is an increment of about 10% in accuracy, but it can be concluded that DBN structure unable to learn from the dataset attributes.

As for "Contraceptive Method Choice" dataset, Evolutionary DBN with bootstrap sampling achieved 98% accuracy as compared to DBN at 47.46% and other algorithms ranges between 42% and 63%, this shows a huge improvement rate. Such result is also shown in "Post-operative" dataset where Evolutionary DBN achieved 94.44% accuracy while other algorithms with exception of DNN, achieved accuracy rate between 72% and 77%. Both "Contraceptive Choice Method" and "Post-operative" has 3 classes, 10 and 9 attributes respectively with different number of instances. Nevertheless, Evolutionary DBN with bootstrap sampling manage to extract the features for the classes and learn from the imbalanced class dataset well and show improvement as compared to DBN in all cases.

### **6.2.3 Weighted Mean Recall**

Table 6.3 presents the weighted mean recall of the algorithms. In Section 4.3.1.4, a recall is an indicator for an algorithm to present the correctly predicted instances out of all the predicted instances. This is explained in Formula 4.2. If the recall rate is

Table 6.3  
Weighted mean Recall of Algorithms

Category	Attributes	Characteristic	Name	Evo-sDBN	DBN	DNN	BPNN	SVM
Binomial	Binary		Hepatitis	<b>1.0</b>	0.81	0.19	0.88	0.88
	Binary		SPECT Heart	<b>1.0</b>	0.83	<b>1.0</b>	0.98	<b>1.0</b>
	Discrete Numerical		SPECTF Heart	<b>0.96</b>	0.83	0.83	0.6	0.76
	Discrete Numerical		Haberman's Survival	<b>0.95</b>	0.6	0.6	0.66	0.73
	Continuous Numerical		Parkinson's Survival	<b>0.97</b>	0.74	0.77	0.18	0.82
	Multivariate		Thoracic Surgery	<b>0.95</b>	0.81	0.81	0.88	0.88
Nominal	Binary		Primary Tumor	<b>1.0</b>	0.54	0.27	0.69	0.77
	Binary		Zoo	0.95	<b>1.0</b>	0.33	0.95	0.9
	Continuous Numerical		Ecoli Protein Localization	<b>1.0</b>	0.68	0.13	0.78	0.76
	Continuous Numerical		Yeast Protein Localization	0.46	0.48	0.29	<b>0.58</b>	0.4
	Multivariate		Standardized Audiology	<b>0.88</b>	0.75	0.33	0.79	0.71
	Discrete Numerical		Contraceptive Method Choice	<b>0.98</b>	0.47	0.46	0.57	0.63
Ordinal	Discrete Numerical		Dermatology	0.6	0.58	0.24	<b>0.94</b>	0.92
	Multivariate		Post-operative	<b>0.94</b>	0.72	0.39	0.72	0.78

closer to 1.0, then it means the algorithm has good recall rate. A recall rate is useful in determining the algorithm is not biased in recalling only the majority class of an imbalanced class dataset, rather also train and test instances from the minority class. An algorithm might have a high accuracy rate, but not a good recall rate. This can mean the algorithm only train and test instances from majority class.

For binomial datasets, Evolutionary DBN with bootstrap sampling possess highest recall rate when compared to other deep learning and machine learning algorithms. In this category, the recall rate for Evolutionary DBN with bootstrap sampling ranges between 0.95 to 1.0, which is very good as opposed to the recall rate from DBN that ranges between 0.6 to 0.83. For other algorithms, the lowest recall rate in binomial category is BPNN for "Parkinson's Survival" dataset at 0.18 and the highest recall rate is SVM for "SPECT Heart" dataset at 1.0. As explained before, it is a possibility that the SVM hyperplane approach is ideal for a dataset with binary attributes and binomial class considering for a similar dataset, "SPECTF Heart", SVM's recall rate is only 0.76. Compare that to Evolutionary DBN with bootstrap sampling, the algorithm achieved recall rate of 1.0 for "SPECT Heart" dataset and 0.96 for "SPECTF Heart" dataset. This shows that Evolutionary DBN with bootstrap sampling manage to learn and recall instances for prediction from both datasets despite its differences in attribute characteristic types and number of attributes as shown in Table 4.1.

For nominal datasets, Evolutionary DBN with bootstrap sampling achieved perfect recall rate of 1.0 for imbalanced class datasets, "Primary Tumor" and "Ecoli Protein Localization". Both imbalanced class datasets has binary and continuous numerical characteristics for their attributes respectively. When we compare the recall rate in each attribute category, Evolutionary DBN with bootstrap sampling has a high recall rate for binary attributes, which scored 1.0 and 0.95 for respective datasets. However, for con-

tinuous numerical attribute category, Evolutionary DBN with bootstrap sampling has a recall rate of 1.0 for "Ecoli Protein Localization" dataset, but only 0.46 for "Yeast Protein Localization" dataset.

As mentioned previously in accuracy rate analysis and according to Tables 4.1 and 4.3, both of the mentioned imbalanced class datasets are similar in attribute characteristics, data distribution and number of classes. Although the difference of number of instances are staggering between the two imbalanced class datasets, it is unlikely that is the main factor for the underperformance of Evolutionary DBN with bootstrap sampling, considering amount of instances for "Yeast Protein Localization" dataset is similar to the amount of instances for "Contraceptive Method Choice" dataset. When compared the performance with other deep learning and machine learning algorithms, the highest recall rate is achieved by BPNN at 0.58, while the rest has recall rate between 0.4 to 0.48. It is fair to conclude that the algorithms have difficulties to learn from this particular imbalanced class dataset. As a conclusion for nominal category, with exception to "Yeast Protein Localization" dataset, Evolutionary DBN with bootstrap sampling scored a high recall rate between 0.88 to 1.0 for the rest of imbalanced class datasets. Although DBN manage to achieve recall rate of 1.0 for "Zoo" dataset, its recall rate in other datasets are fairly low when compared to Evolutionary DBN with bootstrap sampling. Evolutionary DBN with bootstrap sampling manage to show an improvement of recall rate when compared to DBN, which means the algorithm is less biased when using the instances from both majority and minority classes in nominal type imbalanced class datasets.

For ordinal datasets, Evolutionary DBN with bootstrap sampling presents the highest recall rate for two imbalanced class datasets as shown in Table 6.3. As clarified previously in the accuracy rate analysis section, Evolutionary DBN with bootstrap

sampling exhibits low recall rate for "Dermatology" dataset, despite there's an increment from DBN. This shows that the evolutionary and bootstrap sampling feature of the algorithm manage to improve the recall rate of DBN. However, the performance is relatively low when compared to other deep learning and machine learning algorithms. However, the inverse is shown for the other two imbalanced class datasets. For example, for "Contraceptive Method Choice" datasets, Evolutionary DBN with bootstrap sampling exhibits not only the highest recall rate at 0.98, but the difference with the lowest recall rate of DNN at 0.33, is much more as contrast to the comparison between the highest recall rate for "Dermatology" dataset at 0.93 by DNN to 0.6 by Evolutionary DBN with bootstrap sampling. Based on the recall rate for ordinal category, it can be concluded that Evolutionary DBN with bootstrap sampling manage to recall the instances for prediction and also minimize the partiality towards minority classes in the datasets.

#### **6.2.4 Weighted Mean Precision**

Table 6.4 presents the weighted mean precision of the algorithms. Precision is a commonly used performance metric to determine the preciseness of an algorithm. As presented in Formula 4.4, a precision takes account the correctly classified from the actual classification. This measures the sensitivity of an algorithm. The closer its value to 1, the more precise it is. Consistent with accuracy rate and recall rate from Tables 6.2 and 6.3, Evolutionary DBN with bootstrap sampling has the highest precision rate for imbalanced class datasets in binomial category.

However, the precision rate is smaller as compared to recall rate from Table 6.3. This observation can conclude that the algorithm have more false negative (FN) compared to false positive (FP). For example, in "Haberman's Survival" dataset, "0"

Table 6.4  
Weighted mean Precision of Algorithms

Category	Attributes	Characteristic	Name	Evo-sDBN	DBN	DNN	BPNN	SVM
Binomial	Binary		Hepatitis	<b>1.0</b>	0.66	0.04	0.77	0.77
	Binary		SPECT Heart	<b>1.0</b>	0.69	<b>1.0</b>	0.98	<b>1.0</b>
	Discrete Numerical		SPECTF Heart	<b>0.93</b>	0.69	0.69	0.76	0.58
	Discrete Numerical		Haberman's Survival	<b>0.91</b>	0.36	0.36	0.66	0.67
	Continuous Numerical		Parkinson's Survival	<b>0.95</b>	0.55	0.55	0.44	0.67
	Multivariate		Thoracic Surgery	<b>0.9</b>	0.65	0.65	0.78	0.78
Nominal	Binary		Primary Tumor	<b>1.0</b>	0.41	0.07	0.54	0.76
	Binary		Zoo	0.91	<b>1.0</b>	0.11	0.97	0.83
	Continuous Numerical		Ecoli Protein Localization	<b>1.0</b>	0.54	0.02	0.76	0.63
	Continuous Numerical		Yeast Protein Localization	0.21	0.43	0.09	<b>0.59</b>	0.49
	Multivariate		Standardized Audiology	<b>0.8</b>	0.66	0.33	0.79	0.54
	Discrete Numerical		Contraceptive Method Choice	<b>0.96</b>	0.33	0.22	0.57	0.63
Ordinal	Discrete Numerical		Dermatology	0.4	0.48	0.06	<b>0.96</b>	0.95
	Multivariate		Post-operative	<b>0.96</b>	0.52	0.15	0.59	0.6

is labelled when the patient survived 5 years or longer after an operation for breast cancer and "1" is labelled if the patient died within 5 years. Since the FN is higher, it means the number of Type II Error is higher. Basically, the number of patients predicted to survived for 5 years or longer are wrongly classified, when they are actually dead within 5 years are higher, when compared to a smaller number of Type I Error where the number of patients predicted to be dead within 5 years are actually survived for 5 years or longer.

In nominal category, the precision result for all the algorithms is also consistent with accuracy rate and recall rate. Similar to performance of precision rate in binomial category, the precision rate in nominal category is a bit lower as compared to its recall rate. Yet, the precision rate for Evolutionary DBN with bootstrap sampling is high for all the imbalanced class datasets with exception of "Yeast Protein Localization" dataset. The range of precision rate for this category is between 0.09 to 1.0. Evolutionary DBN with bootstrap sampling manage to achieve precision rate between 0.8 to 1.0 for 4 out of 5 imbalanced class datasets.

In ordinal category, Evolutionary DBN with bootstrap sampling achieved the highest precision rate for 2 out of 3 imbalanced class datasets. However, for "Post-operative" dataset, the precision rate is higher than its recall rate. To conclude, the precision rate for Evolutionary DBN with bootstrap sampling is high as it scores between 0.8 to 1.0 except for 2 imbalanced class datasets in both nominal and ordinal categories, which rates at 0.21 and 0.4 respectively. This shows that Evolutionary DBN with bootstrap sampling algorithm is precise for binomial imbalanced class datasets, but for multiclass category, the algorithm performs well but with exception of two imbalanced class datasets as shown in Table 6.4.

### 6.2.5 F1-Score

F1-score is a good measure to decide how synchronized are our recall and precision values. An algorithm can have a good recall rate, but have low precision rate or vice versa. Therefore, it raises the question whether the algorithm has a good performance. This is where F1-score is useful, as it takes into consideration of both recall and precision rates and finds its harmonic value. Similar to recall and precision, as the F1-score is closer to value 1, the algorithm has an agreeable recall and precision rates.

In binomial category, the F1-score achieved by Evolutionary DBN with bootstrap sampling is between 0.92 to 1.0. These scores present an improvement as compared to F1-score achieved by DBN, which is between 0.45 to 0.76. From Table 6.5, SVM has the score between 0.64 to 1.0, which is quite stable as compared to DNN and BPNN. As mentioned previously in accuracy rate, the SVM structure makes it easy to learn and predict from imbalanced class datasets in binomial category. Considering the structure of Evolutionary DBN with bootstrap sampling has similar node structure as DBN, DNN and BPNN, this shows a huge increment of performance in this category. The F1-score observation deduces that Evolutionary DBN with bootstrap sampling has high accuracy as well as high recall and precision.

As for nominal category, with exception of "Yeast Protein Localization", the F1-score achieved by Evolutionary DBN with bootstrap sampling is between 0.84 to 1.0. As we can see from Tables 6.2 and 6.3, although the decrement of accuracy and recall rate between Evolutionary DBN with bootstrap sampling and DBN seems small, due to the low precision value of Evolutionary DBN with bootstrap sampling for the mentioned imbalanced class dataset, the F1-score difference between the algorithm and DBN is large, which is 0.29 and 0.44 respectively. It can be assumed that Evolutionary



Table 6.5  
F1 Score of Algorithms

Category	Attributes	Characteristic	Name	Evo-sDBN	DBN	DNN	BPNN	SVM
Binomial	Binary		Hepatitis	<b>1.0</b>	0.73	0.04	0.82	0.82
	Binary		SPECT Heart	<b>1.0</b>	0.76	<b>1.0</b>	0.98	<b>1.0</b>
	Discrete Numerical		SPECTF Heart	<b>0.94</b>	0.76	0.69	0.64	0.66
	Discrete Numerical		Haberman's Survival	<b>0.93</b>	0.45	0.36	0.66	0.64
	Continuous Numerical		Parkinson's Survival	<b>0.96</b>	0.63	0.55	0.1	0.74
	Multivariate		Thoracic Surgery	<b>0.92</b>	0.72	0.65	0.83	0.83
Nominal	Binary		Primary Tumor	<b>1.0</b>	0.46	0.07	0.59	0.74
	Binary		Zoo	0.93	<b>1.0</b>	0.11	0.95	0.86
	Continuous Numerical		Ecoli Protein Localization	<b>1.0</b>	0.6	0.02	0.75	0.69
	Continuous Numerical		Yeast Protein Localization	0.29	0.44	0.09	<b>0.56</b>	0.33
	Multivariate		Standardized Audiology	<b>0.84</b>	0.7	0.33	0.73	0.61
	Discrete Numerical		Contraceptive Method Choice	<b>0.97</b>	0.37	0.22	0.57	0.63
Ordinal	Discrete Numerical		Dermatology	0.47	0.51	0.06	<b>0.95</b>	0.91
	Multivariate		Post-operative	<b>0.95</b>	0.61	0.15	0.65	0.68

DBN with bootstrap sampling cannot learn from the dataset. However, when compared to other deep learning and machine learning algorithms, "Yeast Protein Localization" dataset only managed to received highest F1-score performance at 0.56 by BPNN. This is very low especially when we contrast to other imbalanced class datasets in nominal category, the highest F1-score can be achieved between 0.84 to 1.0. These high F1-scores are all achieved by Evolutionary DBN with bootstrap sampling. Therefore, despite the previous assumption, it can also be concluded that "Yeast Protein Localization" dataset has a complex feature that is difficult for other deep learning and machine learning algorithms to learn from as well. Nevertheless, for the rest of imbalanced class datasets in nominal category, it can be inferred that Evolutionary DBN with bootstrap sampling's performance for accuracy, recall and precision is established based on its high performance of F1-score.

In ordinal category, Evolutionary DBN with bootstrap sampling has the highest F1-score for 2 out of 3 imbalanced class datasets. It is consistent with its performance for accuracy, recall and precision rates in Tables 6.2, 6.3 and 6.4. In both datasets, Evolutionary DBN with bootstrap sampling shows an improvement when compared to DBN. As for "Dermatology" dataset, the F1-score is affected by the low precision value in Table 6.4. Despite the improved accuracy and recall rates of Evolutionary DBN with bootstrap sampling when compared to DBN, the result shows that the algorithm is not as precise as reflected in other metrics.

#### **6.2.6 Area Under Curve**

Table 6.6 shows the AUC results of the algorithms. As mentioned in Section 4.3.1.6, the AUC is measured between the range of 0 to 1. The closer the result is to 1, it describes the algorithm model has the ability to differentiate between the classes when

Table 6.6  
AUC of Algorithms

Category	Attributes	Characteristic	Name	Evo-sDBN	DBN	DNN	BPNN	SVM
Binomial	Binary		Hepatitis	<b>1.0</b>	0.5	0.5	0.46	0.5
	Binary		SPECT Heart	0.83	0.76	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>
	Discrete Numerical		SPECTF Heart	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	0.48	<b>0.5</b>
	Discrete Numerical		Haberman's Survival	0.5	0.5	0.5	0.47	<b>0.52</b>
	Continuous Numerical		Parkinson's Survival	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>
	Multivariate		Thoracic Surgery	0.5	0.5	<b>0.57</b>	0.5	0.5
Nominal	Binary		Primary Tumor	<b>0.5</b>	0.15	<b>0.5</b>	0.04	0.04
	Binary		Zoo	<b>0.88</b>	0.41	0.5	0.44	0.5
	Continuous Numerical		Ecoli Protein Localization	<b>1.0</b>	0.78	0.5	0.63	0.63
	Continuous Numerical		Yeast Protein Localization	0.5	<b>0.57</b>	0.5	0.48	<b>0.57</b>
	Multivariate		Standardized Audiology	<b>0.94</b>	0.76	0.59	0.42	0.35
	Discrete Numerical		Contraceptive Method Choice	0.5	0.5	0.5	0.53	<b>0.57</b>
Ordinal	Discrete Numerical		Dermatology	<b>0.63</b>	0.58	0.5	0.45	0.48
	Multivariate		Post-operative	<b>0.6</b>	0.5	0.5	0.5	0.5

the algorithm is performing classification tasks.

In binomial category, Evolutionary DBN with bootstrap sampling has a varied outcomes for the imbalanced class datasets. In binary attributes bracket, Evolutionary DBN with bootstrap sampling has good AUC result. DNN, BPNN and SVM achieved highest score of 1.0 for "SPECT Heart" dataset, whereas Evolutionary DBN with bootstrap sampling achieved the second lowest at 0.83. This is consistent with the accuracy rate for this dataset, which means the algorithms manage to learn from the dataset to make accurate classification. However, this perfect score is only achieved for this particular dataset, as it achieved between 0.46 to 0.5 for "Hepatitis" dataset compared to perfect score of 1.0 for Evolutionary DBN with bootstrap sampling.

In discrete and continuous numerical category, Evolutionary DBN with bootstrap sampling achieved 0.5 for all the imbalanced class datasets. The same are achieved by other deep learning algorithms in this category. BPNN achieved the lowest score for both discrete numerical category, while SVM achieved the highest AUC score at 0.52 for "Haberman's Survival" dataset.

The same score of 0.5 is obtained in multivariate attributes characteristic dataset by Evolutionary DBN with bootstrap sampling. DNN achieved the highest AUC score at 0.57 in this regard.

These outcomes exhibit that Evolutionary DBN with bootstrap sampling has a good ability for binary attributes datasets in this category as compared to other deep learning and machine learning algorithms. However, for the rest attribute characteristic categories, Evolutionary DBN with bootstrap sampling have a mediocre performance when predicting the classes from the imbalanced class datasets. This could be attributed to the factor of the the datasets' attribute characteristics. The model could not produce better AUC score when the attributes are not in binary structure.

In nominal category, Evolutionary DBN with bootstrap sampling yields better AUC scores as opposed to other deep learning and machine learning algorithms. For binary attributes, Evolutionary DBN with bootstrap sampling achieved the highest AUC score for both imbalanced class datasets. For "Primary Tumor" dataset, the proposed algorithm achieved score of 0.5, which is not that high but is promising when compared to other deep learning and machine learning algorithms except for DNN which has the same AUC score as the proposed algorithm. The range of AUC score for this imbalanced class dataset is between 0.04 to 0.15. The increment from 0.15 by DBN to 0.5 shows that the Evolutionary DBN with bootstrap sampling improved the performance. "Zoo" dataset obtained the highest AUC score by Evolutionary DBN with bootstrap sampling at 0.88.

For continuous numerical category, Evolutionary DBN with bootstrap sampling achieved the perfect score of 1.0 for "Ecoli Protein Localization" dataset and 0.5 for "Yeast Protein Localization" dataset. AUC score for the proposed algorithm seen a decrement compared to DBN for "Yeast Protein Localization" dataset. This result is consistent with previous performance metrics.

In multivariate attributes characteristic bracket, Evolutionary DBN with bootstrap sampling obtained near perfect score of 0.94 and is the highest when contrasted to other deep learning and machine learning algorithms.

The AUC score for Evolutionary DBN with bootstrap sampling in ordinal category is also satisfactory when compared to other machine learning and deep learning algorithms. For discrete numerical category, Evolutionary DBN with bootstrap sampling achieved 0.5 for "Contraceptive Choice Method" dataset. This result is consistent with deep learning algorithms which are DBN and DNN, whereas machine learning algorithms, BPNN and SVM yield a higher AUC score for this ordinal imbalanced class

dataset.

Evolutionary DBN with bootstrap sampling achieved the highest AUC score for "Post-operative" dataset, which is a multivariate attribute category. The proposed algorithm obtained 0.6 while the other deep learning and machine learning algorithms achieved 0.5 AUC score for this imbalanced class dataset.

Based on the results tabulated, it is concluded that Evolutionary DBN with bootstrap sampling performed better in determining the predicted classes in nominal and ordinal category as compared to the imbalanced class in binomial category. Even though the proposed algorithm does not achieved high AUC score for all imbalanced class datasets in binomial category, it does not have score lower than average.

### **6.2.7 Analysis of Performance Metrics**

Based on the result analyses derived from the precision-recall aspect of performance metrics, Evolutionary DBN with bootstrap sampling outperforms DBN and other deep learning and machine learning algorithms for imbalanced class datasets in binomial category in accuracy rate, weighted mean recall, weighted mean precision and F1-score.

However, for multiclass categories, nominal and ordinal, the result is inconsistent. For example, in nominal category, Evolutionary DBN with bootstrap sampling outperforms DBN and other deep learning and machine learning algorithms for 3 out of 5 imbalanced class datasets.

In "Zoo" dataset, the result for Evolutionary DBN with bootstrap sampling dropped as compared to when DBN is used. Although Evolutionary DBN with bootstrap sampling has fairly high performance, it is likely that the addendum structure of Evolutionary algorithm and bootstrap sampling to the DBN component made the learning from the specific dataset made the weight calculations in the algorithm too complicated when

such dataset is used.

For "Yeast Protein Localization" dataset, the results achieved by all deep learning and machine learning algorithms are low, with the highest accuracy is at 57.91% and F1-score at 0.56 by BPNN. Evolutionary DBN with bootstrap sampling experienced a dropped result of 2% from DBN and even lower F1-score by 0.15 difference compared DBN. Considering the dataset shares the same category and attribute characteristic as "Ecoli Protein Localization", the differences can be found in the number of instances and number of attributes. It can be concluded that the number of instances affects the performance of "Yeast Protein Localization" dataset. On top of that, the difference can be found in the data distribution between the classes. "Yeast Protein Localization" dataset has 10 of total class with the majority is at 31.2% as opposed to the minority class of 0.3%. Other classes distribution is shown in the Table 6.7 below.

Table 6.7  
Yeast Protein Localization data distribution

Class Name	Class distribution
CYT	31.2%
NUC	28.9%
MIT	16.4%
ME3	11.0%
ME2	3.4%
ME1	3.0%
EXC	2.5%
VAC	2.0%
POX	1.3%
ERL	0.3%
<b>Total</b>	<b>100%</b>

The number of class with such distribution can be factored in as a reason Evolutionary DBN with bootstrap sampling and other deep learning and machine learning algorithms performed poorly in the dataset. Compared that to the 336 instances distribution between 8 classes in "Ecoli Protein Localization" dataset presented in Table 6.8 below.

Table 6.8  
Ecoli Protein Localization data distribution

Class Name	Class distribution
cp	42.5%
im	22.9%
pp	15.5%
imU	10.4%
om	6.0%
omL	1.5%
imL	0.6%
imS	0.6%
<b>Total</b>	100%

In ordinal category, Evolutionary DBN with bootstrap sampling performs well for 2 out of 3 imbalanced class datasets. In "Dermatology" dataset, Evolutionary DBN with bootstrap sampling showed an increment of accuracy and recall but showed a decrement in F1-score when compared to DBN. The highest result for this dataset is 94.44% accuracy and 0.95 F1-score by BPNN. SVM also achieved high accuracy rate and F1-score as opposed to other algorithms. This result indicates that machine learning algorithms such as BPNN and SVM performed better than deep learning algorithms for the "Dermatology" dataset.

"Dermatology" dataset has discrete numerical form for its attributes, which is shared with "Contraceptive Method Choice" dataset. The differences between the details of the datasets can be found in the number of instances and number of attributes. In Table 4.1 , "Contraceptive Method Choice" has 1473 instances but belonged to 10 attributes as opposed to in "Dermatology" dataset that has lesser number of instances at 366 and are shared to 34 attributes. This means the learning from a lot of attributes for less number of instances caused Evolutionary DBN with bootstrap sampling unable to perform well. Other than that, the data distribution of the dataset in Table 4.3, the minority class in "Contraceptive Method Choice" is not as low as the distribution for



minority class in "Dermatology" dataset.

It can be concluded that in precision-recall aspect, Evolutionary DBN with bootstrap sampling performs well in binomial category. As for multiclass imbalanced class datasets, Evolutionary DBN with bootstrap sampling performs fairly well even if the result is not as solid as when binomial imbalanced class datasets are used. Another observation that can be made is that Evolutionary DBN with bootstrap sampling performs well on imbalanced class datasets that possess multivariate attributes type in all the categories.

As for AUC aspect, Evolutionary DBN with bootstrap sampling performs fairly well in binomial category although not as concrete as the precision-recall aspect. The proposed algorithm also performs well in both nominal and ordinal categories when handling imbalanced class datasets as the AUC score by Evolutionary DBN is either neutral or performing really well. To conclude, the AUC performance of Evolutionary DBN with bootstrap sampling achieved consistent results when used to predict the imbalanced class datasets utilized in all categories and all characteristic attributes as opposed to the AUC scores by other deep learning and machine learning algorithms that ranges below 0.5 to 1.0.

### **6.3 STATISTICAL TEST ANALYSIS**

This section presents the statistical test analysis performed on the proposed Evolutionary sampled DBN. The result is displayed and discussed as below. The statistical test used for this thesis is the Wilcoxon Signed Rank Test and the Vargha-Delaney A Test. As explained in Sections 4.3.2.1 and 4.3.2.2, both Wilcoxon Signed Rank Test and Vargha-Delaney A Test are non-parametric test. They are useful to test the performance of datasets from multiple domains.

The actual output and predicted output of the Evolutionary DBN with bootstrap sampling are evaluated whether the results achieved in Section 6.2 are established. The result of Wilcoxon Signed Rank Test is presented in Tables 6.3.1, 6.3.1 and 6.3.1, whereas the result of Vargha-Delaney A Test is presented in Table 6.9 respectively. Hypotheses are formulated according to each category and are analysed based on the value achieved by respective statistical test.

### 6.3.1 Wilcoxon Signed Rank Test

For this test, the  $p$  value of the Wilcoxon Signed Rank Test must be more than 0.05 in order to be accepted. For imbalanced class datasets that accomplished accuracy rate of 100%, the hypothesis for such datasets are the differences between the actual output and predicted output must be zero. For other datasets, the assumptions would be the difference between the output must be minimal as Evolutionary DBN with bootstrap sampling can only tolerate minimal mistakes evident to its performance. The hypotheses for binomial datasets are as below:

$H1_0$  : *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Hepatitis dataset*

$H2_0$ : *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for SPECT Heart dataset*

$H3_0$ : *There no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for SPECTF Heart dataset*

$H4_0$ : *There no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Haberman's Survival dataset*

$H5_0$ : *There difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Parkinson's Survival dataset*

$H_{6_0}$ : *There no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Thoracic Surgery dataset*

Dataset	Accuracy	p-value	Annotation
Hepatitis	100%	1.0	Accept
SPECT Heart	100%	1.0	Accept
SPECTF Heart	96.30%	0.15	Accept
Haberman's Survival	95.20%	0.07	Accept
Parkinson's Survival	97.44%	0.35	Accept
Thoracic Surgery	94.70%	0.07	Accept

Table 6.3.1 presents the  $p$  value when actual output and predicted output are compared. The accuracy rate is placed in on of the columns of the table in order to compare whether the accuracy result is justified or happened by chance. For "Hepatitis" and "SPECT Heart" datasets, at 100% accuracy, both datasets achieved  $p$  value of 1.0.

This indicates that both  $H_{1_0}$  and  $H_{2_0}$  are accepted. Since the rest of the imbalanced class datasets in binomial category has accuracy between 94.7% to 97.4%, this means the difference between actual output and predicted output must be minimal. For "SPECTF Heart" and "Parkinson's Survival" datasets, their  $p$  value qualifies for acceptance of  $H_{3_0}$  and  $H_{5_0}$ . As for "Haberman's Survival" and "Thoracic Surgery" datasets, the  $p$  value achieved is 0.07 for both. Although the hypotheses,  $H_{4_0}$  and  $H_{6_0}$  are accepted, the value is close to getting rejected.

Overall, considering  $p$  value achieved by Evolutionary DBN with bootstrap sampling are more than 0.05, the hypotheses for binomial datasets are accepted.

For imbalanced class datasets in nominal category, the hypotheses are formulated the same as for imbalanced class datasets in binomial category. The hypotheses for nominal category are as follows:

H7<sub>0</sub> : *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Primary Tumor dataset*

H8<sub>0</sub>: *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Zoo dataset*

H9<sub>0</sub>: *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Ecoli Protein Localization dataset*

H10<sub>0</sub>: *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Yeast Protein Localization dataset*

H11<sub>0</sub>: *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Standardized Audiology dataset*

<b>Dataset</b>	<b>Accuracy</b>	<b>p-value</b>	<b>Annotation</b>
Primary Tumor	100%	1.0	Accept
Zoo	95.24%	1.0	Accept
Ecoli Protein Localization	100%	1.0	Accept
Yeast Protein Localization	46.13%	$2.2e^{-16}$	Reject
Standardized Audiology	87.5%	0.40	Accept

From Table 6.3.1, "Primary Tumor" and "Ecoli Protein" datasets have perfect  $p$  value of 1.0, which is consistent to their accuracy rate. H7<sub>0</sub> and H9<sub>0</sub> are accepted. "Zoo" dataset has the same  $p$  value of 1.0 even though its accuracy rate is not 100%. However, since the  $p$  value is more than 0.05, H8<sub>0</sub> is accepted. As for "Yeast Protein Localization", Section 6.2 explained that this dataset does not receive high accuracy rate, and this is reflected in its  $p$  value which is less than 0.05. Therefore, the null hypothesis, H9<sub>0</sub> is rejected. "Standardized Audiology" dataset has a  $p$  value of 0.40, which indicates H10<sub>0</sub> is accepted.

H12<sub>0</sub> : *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Contraceptive Method Choice dataset*

H13<sub>0</sub>: *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Dermatology dataset*

H14<sub>0</sub>: *There is no statistical difference between actual output and predicted output for Evolutionary DBN with bootstrap sampling for Post-operative dataset*

Dataset	Accuracy	p-value	Annotation
Contraceptive Method Choice	98.00%	0.8	Accept
Dermatology	68.05%	0.06	Accept
Post-operative	94.44%	1.0	Accept

In ordinal category, the  $p$  value between actual output and predicted output is promising. "Contraceptive Method Choice" dataset has a near perfect accuracy rate, and its  $p$  value is at 0.8. Because it is more than 0.05, so H12<sub>0</sub> is accepted. "Dermatology" dataset's performance is not as thriving as the other two imbalanced class datasets, but because its  $p$  value is at 0.06, it qualifies H13<sub>0</sub> to be accepted. Lastly, although "Post-operative" dataset accuracy rate is not as high as "Comparative Method Choice", its  $p$  value scores at 1.0. This means H14<sub>0</sub> is accepted.

Overall, out of 14 null hypotheses, 13 null hypotheses are accepted and one is rejected.

### 6.3.2 Vargha-Delaney A Test

This section describes the A-measure attained by using Vargha-Delaney A Test. The interpretation of the A-measure is explained in in Section 4.3.2.2. The hypotheses of the experiment is the same as the one in Section 6.3.1. However, the Vargha-Delaney A

Test shows the gap difference value between actual output and predicted output. Table 6.9 details the A-measure for the imbalanced class datasets in each category. Similar to previous tables, the accuracy rate is included to elucidate its consistency with the Vargha-Delaney A Test.

Table 6.9  
The Magnitude of Difference Indicated by A Test Score

Category	Dataset	Accuracy	A-measure	Annotation
Binomial	Hepatitis	100%	0.50	No Difference
	SPECT Heart	100%	0.50	No Difference
	SPECTF Heart	96.30%	0.52	No Difference
	Haberman's Survival	95.20%	0.56	Small Difference
	Parkinson's Survival	97.44%	0.52	No Difference
	Thoracic Surgery	94.70%	0.56	Small Difference
Nominal	Primary Tumor	100%	0.50	No Difference
	Zoo	95.24%	0.50	No Difference
	Ecoli Protein Localization	100%	0.50	No Difference
	Yeast Protein Localization	46.13%	0.78	Large Difference
	Standardized Audiology	87.5%	0.52	No Difference
Ordinal	Contraceptive Method Choice	98.00%	0.52	No Difference
	Dermatology	68.05%	0.58	Small Difference
	Post-operative	94.44%	0.52	No Difference

In binomial category, Section 6.2 explains that Evolutionary DBN with bootstrap sampling performs very well as compared to other categories. For "Hepatitis" and "SPECT Heart" datasets, the A-measure is 0.5 for each. This indicates there is no difference between the actual output and predicted output by Evolutionary DBN with bootstrap sampling for both datasets. Therefore,  $H1_0$  and  $H2_0$  are accepted. "SPECTF Heart" dataset has the accuracy of 96.3%, which is not a perfect accuracy score. However, its A-measure scores at 0.52 indicates that it has no difference. Hence,  $H3_0$  is accepted with no difference. A-measure for "Haberman's Survival" dataset shows that the algorithm has a small difference between the actual output and predicted output.  $H4_0$  is accepted with small difference. "Parkinson's Survival" dataset has a high accuracy rate and is near perfect score. So, its A-measure at 0.52 displays  $H5_0$  is accepted with no difference. Lastly,  $H6_0$  is accepted with small difference because the A-measure is

at 0.56.

For nominal category, "Primary Tumor" and "Ecoli Protein Localization" datasets has A-measure of 0.50 which means  $H7_0$  and  $H9_0$  are accepted with no difference. An interesting observation from Table 6.9 is although "Zoo" does not has 100% accuracy rate, its A-measure is 0.50, which means  $H8_0$  is also accepted with no difference. This finding is consistent with the  $p$  value from Table 6.3.1. "Yeast Protein Localization" dataset has the worst performance by the algorithm as compared to other performance. Its A-measure at 0.78 indicates that for this dataset, the actual output and predicted output has a large difference. Therefore,  $H10_0$  is rejected with a large difference. "Standardized Audiology" dataset has an accuracy rate of 87.5%, which is less than 90%, however its A-measure achieved 0.52 and this portrays there is no difference of comparison between the actual output and predicted output.  $H11_0$  is accepted with no difference.

In ordinal category, both "Contraceptive Method Choice" and "Post-operative" datasets achieved A-measure of 0.52. This observation is congruous with their accuracy rate respectively. Both  $H12_0$  and  $H14_0$  are accepted with no difference between the outputs. Although "Dermatology" dataset has a low performance for its accuracy rate, its A-measure at 0.58 shows that its actual and predicted outputs have small difference with each other.  $H13_0$  is accepted with small difference.

The statistical test carried out by Vargha-Delaney A test is consistent with the results attained by Wilcoxon Signed Rank test. It can be concluded that all the null hypotheses are accepted with exception of one. The actual output and predicted output of Evolutionary DBN with bootstrap sampling has little to no difference when imbalanced class datasets from binomial, nominal and ordinal categories are used.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE WORKS**

#### **7.1 INTRODUCTION**

In this section, the process and results of the experiment is reflected and concluded. Section 7.2 presents the research objectives presented in Section 1.4 and explains whether the objectives are achieved. Section 7.3 describes the limitation of the research and suggestions for future work involving Evolutionary DBN with bootstrap sampling.

#### **7.2 CONCLUSION**

Deep belief network (DBN) exhibits good performance in high level abstraction task. It provides high accuracy and consistent performance in many area, such as speech emotion recognition and semantic segmentation. However, in the case of imbalanced class, the performance of algorithms can be affected and not consistent due to the disproportionate distribution of data between the classes. The first research objective is to study the effects on the performance of DBN when imbalanced class datasets are used for classification task. Literature reviews that study the effects of imbalanced class datasets on machine learning algorithms and DBN is studied and presented in Chapter 2.

Therefore, the second research objective is to propose an optimized DBN model using an evolutionary algorithm with bootstrap sampling for imbalanced class datasets problem. Evolutionary algorithm (EA) is an optimization technique that finds the best optimal solution through fitness test. The best solution is then mutate and recombine to perform better than the previous solution. Bootstrap sampling strategy is also integrated in the algorithm to decrease the inclination to biased data training. Genetic algorithm



(GA) is used as a part of EA to optimize the learning rate, batch size and dropout value of DBN in order to predict the classes better.

The third research objective is to compare the classification performance of optimized DBN algorithm with other machine learning algorithms for imbalanced class datasets. Evolutionary DBN with bootstrap sampling showed a good result as compared to DBN when imbalanced class datasets are utilized. Other deep learning and machine learning algorithms such as deep neural network (DNN), backpropagation neural network (BPNN) and support vector machine (SVM) are also performed to contrast with the performance of Evolutionary DBN with bootstrap sampling. Performance metrics of the algorithm such as accuracy rate, weighted mean recall, weighted mean precision and F1-score are analysed to prove that Evolutionary DBN has the best performance to counter imbalanced class problem.

The last research objective is to analyze the performance of optimized DBN using statistical techniques. Statistical tests such as Wilcoxon Signed Rank test and Vargha Delaney A test are employed to enhance the result of Evolutionary DBN's performance. Hypotheses for each imbalanced class dataset is formed to determine whether the results using performance metrics are established or not. The statistical tests decide the acceptance or rejection of each hypothesis.

Table 7.1  
Research objective achievements

Research Objective	Achievement
To study the effects of DBN algorithm with imbalanced class datasets.	Literature reviews of imbalanced class datasets and its effects on various algorithms are done, including DBN. Imbalanced class datasets are procured and analysed. The performance of DBN algorithm when implementing the imbalanced class datasets are recorded.
To propose an optimized DBN model using an evolutionary algorithm with bootstrap sampling for imbalanced class datasets.	Based on the literature reviews performed for imbalanced class datasets, methods used to minimize the negative effects of imbalanced class datasets on deep learning algorithms are compared and analyzed. The most common approaches for optimization and data sampling is selected and implemented to DBN.
To compare the classification performance of optimized DBN algorithm with other machine learning algorithms for imbalanced class datasets.	Algorithms used for comparison with optimized DBN when imbalanced class datasets are used are DBN, DNN, BPNN and SVM. The imbalanced class datasets are categorised into 3 categories. The classification performance of the algorithms in form of performance metrics such as accuracy rate, weighted mean recall, weighted mean precision and F1-score are recorded and analyzed.
To analyze the performance of optimized DBN using statistical techniques.	Besides performance metrics, the classification performance of optimized DBN is also analyzed using statistical techniques. The techniques used are Wilcoxon Signed Rank test and Vargha Delaney A test. The statistical tests describe the statistical difference between actual output and predicted output and determine the acceptance or rejection of null hypotheses for the experiment.

Evolutionary DBN with bootstrap sampling manage to be the highest performing algorithm for 11 out of 14 datasets. It scored the highest accuracy and F1-score for all datasets in binomial category. Even though Evolutionary DBN with bootstrap sampling saw a decrement for two imbalanced class datasets from DBN in nominal category, it

achieved a high accuracy rate at 95% and F1-score of 0.93 for one of the datasets. Other than that, Evolutionary DBN with bootstrap sampling managed to perform the best in 2 out of 3 imbalanced class datasets in ordinal category.

The statistical tests performed using Wilcoxon Signed Rank test and Vargha-Delaney A test both showed consistent result proving that Evolutionary DBN with bootstrap sampling does improve the DBN model when imbalanced class datasets are used. The difference between the actual and predicted has little to no difference.

To conclude, Evolutionary DBN with bootstrap sampling performs best for binomial category imbalanced class datasets. The imbalanced class datasets with multivariate attribute characteristics for all categories manage to consistently achieve highest performance for the proposed algorithm.

### **7.3 RESEARCH LIMITATION AND FUTURE WORKS**

However, a limitation of the experiment is the inconsistent number of imbalanced class datasets for each category, which are binomial, nominal and ordinal. The reason is because the experiment aims to utilize imbalanced class datasets with disparity that are not modified and is in its original form. This requirement made it difficult to search more imbalanced class datasets for each category due to time constraint.

Another limitation in the experimental setup is the lack of comparison in terms of computational cost. This restraint is due to lack of computation processing power of the hardware employed to execute the algorithm.

For future works, a complete procurement of imbalanced class datasets in each category and elaborate comparison involving computational cost would be ideal. The proposed algorithm can also be implemented for imbalanced class datasets of other domain such as text classification and image classification.

## REFERENCES

- Abdel-Hamid, O., Deng, L., & Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. *Interspeech*.
- Acar, E. (2013). Learning representations for affective video understanding. *MM '13 Proceedings of the 21st ACM international conference on Multimedia*.
- Adler, A., Elad, M., & Zibulevsky, M. (2016). Compressed learning: A deep neural network approach. *arXiv*.
- Al-Fatlawi, A. H., Jabardi, M. H., & Ling, S. H. (2016). Efficient diagnosis system for parkinson's disease using deep belief network. *IEEE*.
- Amato, F., López, A., Peña-Méndez, E. M., Vaňhara, P., Hampl, A., & Havel, J. (2013). Artificial neural networks in medical diagnosis. *Journal of Applied Biomedicine*.
- Amri, A. A., Ismail, A. R., & Zarir, A. A. (2017). Convolutional neural networks and deep belief networks for analysing imbalanced class issue in handwritten dataset. *International Journal on Advanced Science Engineering Information Technology*.
- Amri, A. A., Ismail, A. R., & Zarir, A. A. (2018). Comparative performance of deep learning and machine learning algorithms on imbalanced handwritten data. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*.
- Arora, S., Bhattacharjee, D., Nasipuri, M., Malik, L., Kundu, M., & Basu, D. K. (2010, May). Performance comparison of svm and ann for handwritten devnagari character recognition. *IJCSI International Journal of Computer Science Issues*, 7.
- Asadi, E., da Silva, M. G., Antunes, C. H., is Dias, L., & Glicksman, L. (2014). Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application. *Energy and Buildings*.
- Assodiky, H., Syarif, I., & Badriyah, T. (2017). Deep learning algorithm for arrhythmia detection. *International Electronics Symposium on Knowledge Creation and Intelligent Computing*.
- Aurelio, M., Krizhevsky, R. A., & Hinton, G. E. (2010). Factored 3-way restricted boltzmann machines for modeling natural images. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*.
- Awoyemi, J., Adetunmbi, A., & Oluwadare, S. (2018). Effect of feature ranking on credit card fraud detection: Comparative evaluation of four techniques. *2nd International Conference on Information and Communication Technology and Its Applications (ICTA 2018)*.
- Berry, J., Fasel, I., Fadiga, L., & Archangeli, D. (2012). Training deep nets with imbalanced and unlabeled data. *Interspeech*.

- Bliss, C. A., Frank, M. R., Danforth, C. M., & Dodds, P. S. (2014). An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*.
- Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PLOS ONE*.
- Cao, P., Li, B., Zhao, D., & Zaiane, O. (2013). A novel cost sensitive neural network ensemble for multiclass imbalance data learning. *IEEE*.
- Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligent Research*.
- Chawla, N. V., Japkowicz, N., & Kolcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*.
- Chen, M., Yang, Z., & Liu, W. (2014). Deep neural networks for mandarin tone recognition. *International Joint Conference on Neural Networks (IJCNN)*.
- Chen, Y. (2015). Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE*.
- Ciresan, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks*.
- Ciresan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *IEEE*.
- Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). *Flexible high performance convolutional neural networks for image classification*.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.
- Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013). Improving deep neural networks for lvsr using rectified linear units and dropout. *IEEE*.
- Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and roc curves. *Proceedings of the 23rd International Conference on Machine Learning*.
- Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7.
- Deng, L. (2012). Three classes of deep learning architectures and their applications: A tutorial survey. *APSIPA transactions on signal and information processing*.
- Deng, L., Hinton, G., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. *IEEE*.
- Deshmukh, A. R., & Akarte, P. S. P. (2016). Classification of imbalanced medical data efficiently using multiclass svm and genetic algorithm. *International Journal of Computer Science and Mobile Computing*.

- Dieleman, S., mon Brakel, P., & Schrauwen, B. (2011). Audio-based music classification with a pretrained convolutional network. *ISMIR*.
- Drew, P. J., & Monson, J. R. T. (2000, January). Artificial neural networks. *Surgery*.
- Elhoseny, M., Yuan, X., Yu, Z., Mao, C., El-Minir, H. K., & Riad, A. M. (2015). Balancing energy consumption in heterogeneous wireless sensor networks using genetic algorithm. *IEEE*.
- Fernandez, A., Garcia, S., & Herrera, F. (2011). Addressing the classification with imbalanced data: Open problems and new challenges on class distribution. *Springer-Verlag Berlin Heidelberg*.
- Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4).
- Ganguly, S., & Samajpati, D. (2015). Distributed generation allocation on radial distribution networks under uncertainties of load and generation using genetic algorithm. *IEEE*.
- Ghahabi, O., & Hernando, J. (2014). Deep belief networks for i-vector based speaker recognition. *EEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*.
- Ghamisi, P., & Benediktsson, J. A. (2007). Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *JOURNAL OF LATEX CLASS FILES*, 6(1).
- Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. *Springer-Verlag Berlin Heidelberg*.
- Han, K., Yu, D., & Tashev, I. (2014a). Speech emotion recognition using deep neural network and extreme learning machine. *Interspeech*.
- Han, K., Yu, D., & Tashev, I. (2014b). Speech emotion recognition using deep neural network and extreme learning machine. *Interspeech*.
- Haque, M. N., Noman, N., Berretta, R., & Moscato, P. (2016). Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification. *Plos One*.
- Hensman, P., & Masko, D. (2015). *The impact of imbalanced training data for convolutional neural networks* (Unpublished doctoral dissertation). KTH Royal Institute of Technology.
- Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8), 1711–1800.
- Hinton, G. E. (2007). Learning multiple layers of representation. *Elsevier Ltd*.
- Hong, S., Noh, H., & Han, B. (2015). Decoupled deep neural network for semi-supervised semantic segmentation. *Advances in Neural Information Processing Systems*.

- Honglak Lee, R. R., Roger Grosse, & Ng, A. Y. (2011). Unsupervised learning of hierarchical representations with convolutional deep belief networks. *communications of the acm*.
- Huang, C., Gong, W., Fu, W., & Feng, D. (2014). A research of speech emotion recognition based on deep belief network and svm. *Hindawi Publishing Corporation*.
- Huang, J.-T., Li, J., Yu, D., Deng, L., & Gong, Y. (2013). Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. *IEEE*.
- Hwang, J. P., Park, S., & Kim, E. (2011). A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function. *Elsevier Ltd*, 8580–8585.
- Inanlo, E., & Zadeh, J. M. (2016). Social networks classification using dbn neural network based on genetic algorithm. *International Journal of Innovations Advancement in Computer Science*.
- Jamshidi, M., Ghaedi, M., Dashtian, K., Hajatib, S., & Bazrafshana, A. (2015). Ultrasound-assisted removal of al<sup>3+</sup> ions and alizarin red s by activated carbon engrafted with ag nanoparticles: central composite design and genetic algorithm optimization. *RSC Advances*.
- Jang, H., Plis, S. M., Calhoun, V. D., & Lee, J.-H. (2017). Task-specific feature extraction and classification of fmri volumes using a deep neural network initialized with a deep belief network: Evaluation using sensorimotor tasks. *Elsevier Ltd*.
- Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y., & Guan, R. (2016). Text classification based on deep belief network and softmax regression. *CrossMark*.
- Jo, T., & Lee, J.-H. (2015). Latent keyphrase extraction using deep belief networks. *International Journal of Fuzzy Logic and Intelligent Systems*.
- J.Pradeep, E.Srinivasan, & S.Himavathi. (2011). Diagonal based feature extraction for handwritten character recognition system using neural network. *IEEE*.
- Kaur, G., Srivastava, M., & Kumar, A. (2018). Genetic algorithm for combined speaker and speech recognition using deep neural networks. *Journal of Telecommunications and Information Technology*.
- Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2016). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE TRANSACTIONS ON NEURAL NETWORKS*.
- Khandelwal, D., Schoukens, M., & th, R. T. (2018). Grammar-based representation and identification of dynamical systems. *arXiv*.
- Koturwar, P., Girase, S., & Mukhopadhyay, D. (2015). A survey of classification techniques in the area of big data. *arXiv preprint arXiv:1503.07477*.
- Kuang, D., & He, L. (2014). Classification on adhd with deep learning. *International Conference on Cloud Computing and Big Data*.

- Kuremoto, T., Kimura, S., Kobayashi, K., & Obayashi, M. (2014). Time series forecasting using a deep belief network with restricted boltzmann machines. *Elsevier Ltd.*
- Lanbouri, Z., & Achchab, S. (2015). A hybrid deep belief network approach for financial distress prediction. *IEEE*.
- Le, D., & Provost, E. M. (2013). Emotion recognition from spontaneous speech using hidden markov models with deep belief networks. *IEEE*.
- Liu, D., Niu, D., Wang, H., & Fan, L. (2014). Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm. *Elsevier Ltd.*
- Liu, W., & Chaw, S. (2011). Class confidence weighted knn algorithms for imbalanced data sets. *Springer*.
- Liu, Y., Yu, X., Huang, J. X., & An, A. (2010). Combining integrated sampling with svm ensembles for learning from imbalanced datasets. *Elsevier Ltd.*
- Lopes, N., Ribeiro, B., & o Gonc alves, J. (2012). Restricted boltzmann machines and deep belief networks on multi-core processors. *IEEE World Congress on Computational Intelligence*.
- López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences* 250.
- Ma, L. L., & Wu, J. (2014). A tibetan component representation learning method for online handwritten tibetan character recognition. *14th International Conference on Frontiers in Handwriting Recognition*.
- Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Elsevier Science Ltd.*
- Megumi, F., Yamashita, A., Kawato, M., & Imamizu, H. (2015). Functional mri neurofeedback training on connectivity between two regions induces long-lasting changes in intrinsic functional network. *Frontiers in Human Neuroscience*.
- Mirjalilia, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software* 95.
- Mohamed, A., Dahl, G. E., & Hinton, G. (2012, January). Acoustic modeling using deep belief networks. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 20(1), 14-22.
- Mohamed, A., Hinton, G., & Penn, G. (2012). Understanding how deep belief networks perform acoustic modelling. *ICASSP*.
- Mohamed, A., Yu, D., & Deng, L. (2010). Investigation of full-sequence training of deep belief networks for speech recognition. *Interspeech*.



- Mrazova, I., & Kukacka, M. (2012). Can deep neural networks discover meaningful pattern features? *Elsevier Ltd*, 194 – 199.
- Neath, M. J., Swain, A. K., Madawala, U. K., & Thrimawithana, D. J. (2014). An optimal pid controller for a bidirectional inductive power transfer system using multiobjective genetic algorithm. *IEEE TRANSACTIONS ON POWER ELECTRONICS*.
- Niu, X. X., & Suen, C. Y. (2011, October). A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Elsevier Ltd*, 1318–1325.
- Oh, S.-H. (2013). Protein disorder prediction using multilayer perceptrons. *International Journal of Contents*.
- Oreski, S., & Oreski, G. (2014). Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Elsevier Ltd*.
- Perry, T., Bader-El-Den, M., & Cooper, S. (2015). Imbalanced classification using genetically optimized cost sensitive classifiers. *IEEE*.
- Qiong, G., Li, Z., & Zhihua, C. (2009). Evaluation measures of the classification performance of imbalanced data set. *Proceedings of the 4th International Symposium ISICA*.
- Qiu, M., Ming, Z., Li, J., Gai, K., & Zong, Z. (2015). Phase-change memory optimization for green cloud with genetic algorithm. *IEEE*.
- Ren, J. (2012). Ann vs. svm: Which one performs better in classification of mccs in mammogram imaging. *Elsevier Ltd*.
- Richardson, F., Reynolds, D., & Dehak, N. (2015). A unified deep neural network for speaker and language recognition. *arXiv*.
- Rosca, M. C. (2014). *Networks with emotions an investigation into deep belief nets and emotion recognition* (Tech. Rep.). Imperial College London.
- Sainath, T. N., Kingsbury, B., Sindhvani, V., Arisoy, E., & Ramabhadran, B. (2013). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. *IEEE*.
- Sazal, M. M. R., Biswas, S. K., Amin, M. F., & Murase, K. (2013). Bangla handwritten character recognition using deep belief network. *International Conference on Electrical Information and Communication Technology*.
- Siniscalchi, S. M., Yu, D., Deng, L., & Lee, C.-H. (2013). Exploiting deep neural networks for detection-based speech recognition. *Neurocomputing 106*.
- Sun, Y., Wang, X., & Tang, X. (2013). Hybrid deep learning for face verification. *IEEE International Conference on Computer Vision*.
- Sun, Z., Sun, H., & Zhang, J. (2018). Multistep wind speed and wind power prediction based on a predictive deep belief network and an optimized random forest. *Hindawi Mathematical Problems in Engineering*.

- Swersky, K., Chen, B., Marlin, B., & de Freitas, N. (2010). A tutorial on stochastic approximation algorithms for training restricted boltzmann machines and deep belief nets. *IEEE*.
- Tang, Y., Zhang, Y.-Q., Chawla, N. V., & Krasser, S. (2009). Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems*.
- Terusaki, K., & Stigliani, V. (2014). Emotion detection using deep belief networks.
- Uci machine learning*. (2018). Retrieved 2018-10-01, from <https://archive.ics.uci.edu/ml/datasets.php>
- Vargha, A., & Delaney, H. D. (2000). A critique and improvement of the cl common language effect size statistics of mcgraw and wong. *Journal of Educational and Behavioral Statistics*.
- Walid, R., & Lasfar, A. (2014). Handwritten digit recognition using sparse deep architectures. *IEEE*.
- Wang, T., W, D. J., Coates, A., & Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. *ICPR*.
- Weiss, G. M., & Provost, F. (2001). The effect of class distribution on classifier learning: An empirical study. *Technical Report ML-TR-44*.
- Xie, J., Lu, H., Nan, D., & Nengbin, C. (2010). Sparse deep belief net for handwritten digits classification. *Springer-Verlag Berlin Heidelberg*, 71–78.
- Yan, Y., Chen, M., Shyu, M.-L., & Chen, S.-C. (2015). Deep learning for imbalanced multimedia data classification. *IEEE International Symposium on Multimedia*.
- Yu, L., Yang, Z., & Tang, L. (2015). A novel multistage deep belief network based extreme learning machine ensemble learning paradigm for credit risk assessment. *Springer Science+Business Media New York*.
- Zhai, J., Zhang, S., & Wang, C. (2015). The classification of imbalanced large data sets based on mapreduce and ensemble of elm classifiers. *Springer-Verlag Berlin Heidelberg*.
- Zhang, C., & Tan, K. C. (2016). Training cost-sensitive deep belief networks on imbalance data problems. *IEEE*.
- Zhang, C., Tan, K. C., Li, H., & Hong, G. S. (2018, May). A cost-sensitive deep belief network for imbalanced classification. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*.
- Zhang, D., & Xu, W. (2014). A data-distribution based imbalanced data classification method for credit scoring using neural networks. *IEEE*.
- Zhang, X., Trmal, J., Povey, D., & Khudanpur, S. (2014). Improving deep neural network acoustic models using generalized maxout networks. *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*.
- Zhang, Y., Wang, S., Phillips, P., & Ji, G. (2014). Binary pso with mutation operator for feature selection using decision tree applied to spam detection. *Elsevier Ltd.*

- Zhao, Z., Jiao, L., Zhao, J., Gu, J., & Zhao, J. (2017). Discriminant deep belief network for high-resolution sar image classification. *Elsevier Ltd.*
- Zhao, Z., Yang, Z., Luo, L., Lin, H., & Wang, J. (2016). Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics.*
- Zheng, W. L., Zhu, J. Y., Peng, Y., & Lu, B. L. (2014). Eeg-based emotion classification using deep belief networks. *Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Eng..*
- Zou, S., Huang, Y., Wang, Y., Wang, J., & Zhou, C. (2008). Svm learning from imbalanced data by ga sampling for protein domain prediction. *IEEE Computer Society.*