南華大學科技學院永續綠色科技碩士學位學程

碩士論文

Master Program of Green Technology for Sustainability
College of Science and Technology
Nanhua University

Master Thesis

應用數據挖掘技術於921震災崩塌地影像分類之比較

A Comparison of Chi-Chi Earthquake-induced Landslide Detection in Central Taiwan using Data Mining Techniques

舒堤杰

Sumriti Ranjan Patra

指導教授：林文賜 博士

Advisor: Wen-Tzu Lin, Ph.D.

中華民國 110 年 5 月

May 2021

南 華 大 學

永續綠色科技碩士學位學程

碩 士 學 位 論 文

應用數據挖掘技術於921震災崩塌地影像分類之比較

A Comparison of Chi-Chi Earthquake-induced Landslide detection in Central Taiwan using Data Mining Techniques

研究生：Sumrit Ranjan Patra (舒堤杰)

經 考 試 合 格 特 此 證 明

口試委員：林昭遠

林文斌

李長元

指導教授：林文斌

系主任(所長)：林文斌

口試日期：中華民國 110 年 5 月 21 日

ACKNOWLEDGEMENT

First and foremost, I would like to thank my thesis research supervisor Dr. Wen-Tzu Lin, Director, Master Program of Green Technology for Sustainability, Nanhua University, Dalin Township, Chiayi County, Taiwan for his constant guidance and support throughout the course of this research. I am also sincerely thankful to Mrs. Liang, our department assistant for her constant involvement and for being there whenever I needed her for any documentation work.

My sincere appreciations also go to Dr. Hong Yao-Ming, Professor at the Department of Green Technology for Sustainability for his course on Data Mining which was highly beneficial and introduced me to a new field, without it this research would have been hardly possible.

I especially owe the sincerest thanks to my cousin Dr. Lokanath Patra, Postdoctoral Scholar, Department of Physics, Michigan Technological University, U.S.A for his deep involvement and guidance throughout the course of writing each section in this thesis. Also, I would sincerely like to thank my parents for their constant encouragement to pursue my goals and their belief in my potential.

My sincere thanks to the Ministry of Education (M.O.E), Government of Taiwan (R.O.C) for allowing me to pursue higher education in Taiwan which was a wonderful experience.

Finally, I am deeply indebted to Nanhua University for allowing me to pursue a Master's degree here in Taiwan.

Author

Sumriti Ranjan Patra

Master Program of Green Technology for Sustainability, Nanhua University

中文摘要

全世界大部分國家都曾經遭遇崩塌地引發之後果，其形式包括經濟衰退、數十億美元的損失和更多的傷亡。崩塌地成因有先天性的自然因素或裸露的陡峭斜坡沖刷而形成，或者是由自然災害引起的如地震，火山爆發，積雪融化和豪雨等。氣候變遷是一個全球性挑戰，其極端氣候的暴雨，對有些易引發大規模崩塌地區，將面臨災難性後果。因此這些挑戰急需發展一套快速有效的評估系統，能產生錯誤率較低之崩塌分類成果，有效且精確判釋出崩塌區位與非崩塌區位供後續坡地防災應用。

台灣在 1999 年 9 月 21 日發生芮氏 7.3 級地震，造成灣中部地區山坡地大規模的崩塌裸露，對該地區之經濟、聚落、設施和生態系統造成嚴重破壞。本研究以受災較嚴重之九九峰地區為試區，透過資料探勘技術對震災後 SPOT 影像資料進行判釋分析，探討崩塌區位判釋之最佳方法。本研究評估九種資料探勘方法，首先以兩個光譜指數 NDVI 及 TBI 來萃取崩塌與非崩塌區位之光譜影像值，並建立其判釋模式萃取崩塌區位，並導入九種資料探勘方法，包括四個非監督分類法如 K-means、Minibatch K-means、BIRCH 和 GMM 等，以及 5 個監督分類法如 SVM、DT、ET、RF 和 XGBoost 等，進行崩塌分類成果及精度比較，並計算出各種定量統計數據以供驗證，包括總體精確度(OA)、Kappa 值、使用者精度(UA)及生產者精度(PA)等。

研究結果指出 K-means 方法優於其他分類方法，其總體精確度為 95.94%，相較於 XGBoost 為 95.49％、DT 和 RF 為 95.38％，而 SVM 方法之總體精確度最低為 92.51％。整體而言，這九種方法都有還不錯的表現，總體精確度皆達到 90％以上，表示其在判釋崩塌地上具重要意義。而未來

對於崩塌區位之判釋應用上，建議可結合類神經之深度學習演算法或加入其他地形因子以有效提升判釋之精確度。

關鍵詞: 氣候變化、崩塌區位、光譜指數、資料探勘技術

ABSTRACT

Major countries worldwide face the consequences of landslides in the form of reduced economy, damages worth billions, and higher fatalities. Landslides are the result of upsetting steeper inclines that were previously preconditioned or bare of vegetation. They are caused by natural hazards such as earthquakes, volcanic eruptions, melting of snow, and heavy rainfall showers.

Climate change represents a global challenge that induces frequent volcanic and seismic activities due to tectonic excitation, and torrential rainfall with an abnormal downpour. Some regions are not preconditioned to tolerate such extreme weather changes and face cataclysmic repercussions in the form of landslides. Such challenges call for urgent development of a fast and efficient assessment system generating error-free Landslide Inventory Maps (LIM) that depicts a clear boundary between affected and unaffected regions.

Taiwan is one such country that has been enduring such calamities throughout recent years. One particular catastrophic incident in the form of an earthquake having a magnitude of 7.3 in Richter scales which occurred on 21st September 1999, devastated the central part of Taiwan inflicting serious damages to its economy, human livelihood, infrastructure, and ecosystem. Mt Jou-Jou one of the severely impacted regions was adopted for this study. Analysis of Geo-Spatial data with data mining is cost-efficient and reduces dangerous fieldwork.

This research explores the potential of nine data mining techniques along with a pixel-based image differencing on two spectral indices i.e., Normalized Difference Vegetation Index (NDVI) and Total Brightness Index (TBI) derived from multi-temporal SPOT satellite imagery for landslide detection. Landslide maps were generated and compared from four unsupervised i.e., K-means, Minibatch K-means,

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), and Gaussian Mixture Models (GMM), and five supervised learning algorithms which included Support Vector Machines (SVM), Decision Tree (DT), Extra Trees (ET), Random Forest (RF), and Extreme Gradient Boosting (XGBoost). For comparison, a validation set was employed from which various quantitative statistics were computed such as Overall Accuracy (OA), Kappa Statistics (K), User's Accuracy (UA), and Producer's Accuracy (PA).

The results suggested that the K-means algorithm outperformed other algorithms and showed the highest overall accuracy of 95.94% with a close follow-up from XGBoost (95.49%), DT, and RF (95.38%). The lowest accuracy was yielded by the SVM algorithm of 92.51%. In general, all the algorithms delivered outstanding performance and achieved overall accuracies well above 90% indicating their significance for identifying landslides.

The overview of the conclusion in this research stated that landslide mapping using data mining algorithms and SPOT-derived spectral indices can provide essential surface information that can further act as an efficient tool for future landslide detection problems & research such as comparison with other deep learning algorithms and further addition of topographic, geologic, morphologic and lithologic information to the dataset. However, the proposed system in this research can be further used in regions with similar topographic and geologic nature.

Keywords: Climate Change, Landslide Inventory Map, Spectral Index, Data Mining

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

BIRCH	Balanced Iterative Reduced & Clustering using Hierarchies
GMM	Gaussian Mixture Models
SVM	Support Vector Machines
DT	Decision Tree
ET	Extra Trees
RF	Random Forest
XGBoost	Extreme Gradient Boosting
NDVI	Normalized Difference Vegetation Index
TBI	Total Brightness Index
LIM	Landslide Inventory Map
RMSE	Root Mean Square Error
OA	Overall Accuracy
K	Kappa Statistics
PA	Producer's Accuracy
UA	User's Accuracy
CNN	Convolution Neural Networks
ANN	Artificial Neural Networks

Chapter 1 Preface

1.1 Introduction

Landslides are considered as a third most significant natural hazard that has distressed several countries including loss of lives and damages costing billions of dollars worldwide. They are defined as the mass movement of debris, rocks, and soil in large volumes, or slope failures. Some triggering factors include heavy rainfall, earthquakes, volcanic eruptions, snowmelt, etc. Sometimes, landslides are also caused by man-made disturbances such as quarrying, construction, unplanned landscape changes, and urban expansion in addition to natural factors (Tran et al., 2019). Their existence can also be ascribed by the geological, and meteorological processes on earth such as lithology, and slope morphology (Ma et al., 2020). Often landslides are the result of rocks, soil, and slope preordained to fail. They are considered a global threat that endangers human lives and causes several disruptions by substantial damages to highways, bridges, buildings, and other human-inhabited areas. Therefore, the requirement of an early warning system backed by accurate landslide susceptibility mapping is imminent. However, susceptibility mapping first requires prior information on past landslide events that occurred within the last 10 to 15 years and their triggering factors. For this purpose, a comprehensive Landslide Inventory Map with precise landslide spatial signatures and distribution is necessary.

Landslide identification or detection is the process of accurately delineating the boundary between landslides and non-landslide regions. These regions are usually depicted in a map which is generally known as a Landslide Inventory Map (LIM) which includes potential and comprehensive landslide patterns. Landslide identification plays a crucial role in disaster risk assessment and management (Danneels, 2007; Chen et al., 2014; Sameen et al., 2019; Wang et al., 2020). The

landslide inventory requires regular updates following major climatic events such as heavy rainfall, seismic and volcanic events, etc. Thus, such information should be regularly supplied to decision-makers for policy-making and mitigation strategies.

Landslide identification is conventionally conducted through visual interpretation of aerial photographs which requires field inspection that consumes time, costly, labor-intensive, and sometimes dangerous or inaccessible (Lei et al., 2018; Pradhan, 2018; Sameen, 2019; Tran et al., 2019). Occasionally, this technique fails to detect small-scale failures which could increase the false negatives in the resulting generated LIM. In this context, integrating remote sensing data in a comprehensive framework of landslide detection is cost-effective, due to its wide availability, accessibility, and coverage which contributes to the rapid up-gradation of landslide inventory. Several remote sensing satellite sensors are available in the market that provides images with large coverage along with incredible spatial resolution. Some of the dominant remote sensing sensors are "Optical Remote Sensing and Synthetic Aperture Radar" (SAR), "Airborne Laser Scanning" (LiDAR), "Satellite Pour I 'Observation de la Terre" (SPOT), "Advanced Spaceborne Thermal Emission, Reflection Radiometer" (ASTER), "Interferometry Synthetic Aperture Radar" (InSAR), "RapidEye", "Unmanned Aerial Vehicle" (UAV), and other sensors include "Landsat", "IKONOS", "Quickbird". However, directly implementing raw remote sensing data without prior preprocessing or suitable feature subset extraction and selection could severely diminish the accuracy of the resulting landslide inventory map. Besides, the proper integration of techniques that develops decision boundaries or rules between these features and the landslides is also vital. On this basis, incorporating data mining techniques with remote sensing data could further contribute to the advancements of landslide detection technology and produce a more sophisticated framework for future studies.

In addition to remote sensing and data mining, some image analysis methods are also incorporated in landslides studies: “Pixel Based Image Analysis” (PBIA) and “Object-Based Image Analysis” (OBIA). Pixel-based methods only have a single-pixel as their core processing element in which image correlation is commonly implemented such as image stacking or image differencing. Object-based also depends on single pixels as well as an image object which is composed of pixels having similar spectral signatures (Sameen et al., 2019), same as clusters. One of the widely implemented techniques in landslide studies is image segmentation (Li et al., 2015; Keyport, 2018; Tavakkoli et al., 2019). Unlike PBIA, the OBIA can be implemented on multiple scales (Pradhan et al., 2018).

In general, the integration of suitable remote sensing image analysis, feature selection, and data mining techniques can configure a thorough landslides analysis and produce precise landslide inventory maps. These data mining techniques are categorized into unsupervised and supervised approaches: the former requires no prior training whereas the latter needs pre-defined labeled data. In recent years, numerous geologic researchers incorporated such frameworks with comprehensive comparisons and produced highly accurate landslide maps. Some studies included feature selection along with the machine learning model’s comparison. Wang et. al. (2020) integrated topographic, rainfall, geologic, lithological factors to compare CNN, SVM, and RF models. Another study featured factor importance and compared several types of decision tree models (Alkhasawneh et al., 2014). Pradhan (2018) produced a comparison of six feature selection techniques using LiDAR point cloud orthophotos and their effects on classification using SVM and RF. A detailed feature quantitative statistical analysis was proposed by Chen et. al. (2014) on LiDAR-derived Digital Terrain Model (DTM) with RF model. Sameen & Pradhan

(2019) proposed a fusion of spectral and topographic information to compare deep learning models for landslide detection.

Predictive systems based on models by machine learning (data mining) are under constrained. For example, models that show superior performance on a particular dataset may diverge on circumstances and datasets outside their local area due to the complex earth system (Ma et al., 2020). Hence, landslide detection based on data mining faces various challenges from the uncertain and complex dataset. Consequently, Ghorbanzadeh et. al. (2019) investigated the effects of data augmentation to artificially increase training samples on deep learning and machine learning models. Some studies also conducted a detailed comparison between pixel-based and object-based approaches for landslide detection. Li et. al. (2015) comprehensively compared them and investigated their sensitivity to feature selection and suggested a semi-automated technique with OBIA for forested landslide identifications. Few studies have also compared PBIA and OBIA along with impacts of initially defined clusters on unsupervised algorithms (Keyport et al., 2018; Tran et al., 2019). Image segmentation-based object analysis was performed on both single and multi-scale for fuzzy classification and stacking of machine learning models together by Lei et. al. (2018) & Tavakkoli et. al. (2019), respectively.

Taiwan is a small island nation located on the East Asia coast and west of the Pacific (Tsai et al., 2010). This country is regularly distressed by natural hazards such as earthquakes, torrential rainfalls, forest fires, and typhoons (Lin et al., 2004). These hazards are the prime reasons for inducing landslides in its mountainous region. These regions have rugged terrains and are believed to have delicate geologic conditions that result in severe landslides and floods due to heavy precipitation (Tsai et al., 2010). Past research works in Taiwan mainly focused on analyzing and

providing adequate information about the landslide and their contributing factors. The research works continuously strive to update the landslide inventory and provide accurate info on damaged areas to local authorities for remedial strategies and programs for landslide prevention. Tsai et. al. (2010) produced a detailed landslide map and analyzed its contributing factors in southern Taiwan following the devastating Typhoon Morakot in 2009.

On 21st September 1999, Taiwan was devastated by a cataclysmic earthquake that severely damaged the ecosystem, geographic terrain, and human lives. The epicenter was Jiji (Chi-Chi) Township, Nantou County, Central Taiwan. The earthquake was named after its epicenter and date of occurrence as Chi-Chi or 921-earthquake, respectively. It was considered the second-deadliest earthquake in the history of Taiwan after the Shichiku-Taichu Earthquake in 1935. The seismic event during the 921-earthquake reached the Richter scale of 7.3 at the center (Lin et al., 2004; Lin et al., 2006; Lin, 2008; Yang et al., 2017). The earthquake-induced severe landslides representing a large movement of stones and rocks that turned into debris. Under such circumstances damage to properties, utilities, crops, and danger to human lives is inevitable. The Chi-Chi earthquake caused several difficulties including dammed lakes and high casualty. Several studies were conducted following the seismic event to provide adequate information on the damages. Wan (2010) conducted a landslide study and modeling using data mining techniques in Shei-Pa National Park which was a major source of energy for Central Taiwan. Another landslide area i.e., Chiufenershan was monitored for vegetation restoration following the 921 earthquake-induced landslides (Lin et al., 2008). However, one of the severely damaged hillslopes due to landslides was on Mt Jou-Jou. Majority of the upper hillside was bare of vegetation that resulted in a dramatic rise of suspended solids in the air that severely reduced its quality. Hence, the local authorities

announced Mt Jou-Jou as a natural reservation park for landslide and restoration purposes. Mt Jou-Jou has been a case study in several past research works following the earthquake. Lin (2004) found out that the vegetation recovery rate after one year of the earthquake was 47.1% and the vegetation succession reached 89.69% after six years (Lin et. al. 2006). However, a detailed study was conducted by Yang et. al. (2017) conducted an extended vegetation assessment using 14 SPOT satellite images and pointed out that short-term observation and omission of seasonal variations could provide inadequate information on actual vegetation status. Hence, past research works in Taiwan showed vegetation condition as one of the most prominent as well as a preventive factor on landslides.

1.2 Research Objectives

The primary objective of this study is to investigate and develop an efficient comparative framework on some of the unexplored algorithms with widely adopted ones to detect landslides from remote sensing data. To accomplish this, specific objectives of the present study are stated below:

(1) To establish a comprehensive framework of four unsupervised and five supervised methods to identify the topographic signatures of landslides and non-landslide terrains using SPOT-derived spectral index. Specifically, the unsupervised approach includes K-means, Minibatch K-means, Gaussian Mixture Models (GMM), and Balanced Iterative and Reducing Clustering using Hierarchies (BIRCH) whereas the supervised approach mainly comprises of Support Vector Machines (SVM), Decision Tree (DT), Extra Trees (ET), Random Forest (RF), and Extreme Gradient Boosting (XGBoost).

(2) To study the implementation of pixel-based image differencing method from bi-temporal images to highlight the landscape undergoing drastic changes post-natural hazard.

(3) To analyze the characteristics of features such as Normalized Difference Vegetation Index (NDVI) and Total Brightness Index (TBI) simultaneously in locating the landslide areas.

(4) To compare nine generated landslide maps and determine the best algorithm for landslide inventory.

1.3 Research Framework

This study includes a novel approach to compare nine data mining algorithms for landslide detection which are subdivided into four unsupervised and five supervised algorithms. The study first includes the preprocessing of the raw spectral image to reduce contrast and noise in the pixels. Features derived from the digital values of the pixels are extracted. Due to the multi-temporal nature of the images, pixel-level image differencing was conducted to visualize regions undergoing drastic change post-earthquake. The unsupervised approach doesn't require a pre-labeled dataset; hence, clustering was directly conducted on all the pixels. The resulting clusters were manually appointed to landslide (L) and non-landslide (NL) groups. The supervised algorithms were precisely trained on the extracted dataset from the differenced images. The major drawback is the ill effects on their performance by imbalance and noise in the dataset. These effects are handled through various external techniques and in-built features inside the algorithms that are widely known in the data mining community. The model overfit and underfit is also monitored before its deployment for landslide identification. The generated landslide maps from all the algorithms were quantitatively compared based on several statistical values. The significance of the correlation between features is also established for landslide assessment through this research.

The remainder of the thesis is structured as follows: Chapter 2 introduces a detailed review of the past research works related to landslides and the proposed

study area. Chapter 3 includes a detailed geographic and geologic description of the study area followed by a detailed discussion on the theory, mathematics, and working principles behind each algorithm along with the proposed framework which includes integration of remote sensing data, its preprocessing, clustering, supervised training methodology, and quantitative comparative tools for the generated landslide maps has also been presented.

The comparative results obtained are presented in Chapter 4 in the form of quantitative statistical analysis through validation sets for each generated Landslide Inventory Map. In addition to this, the training of each supervised algorithm was checked based on overfitting and underfitting through benchmarking methods.

Finally, Chapter 5 includes an interpretation of the acquired results along with a detailed discussion on their significance concerning the past research works. A brief conclusion to this study as well as suggestions for future researches is also illustrated in this chapter.

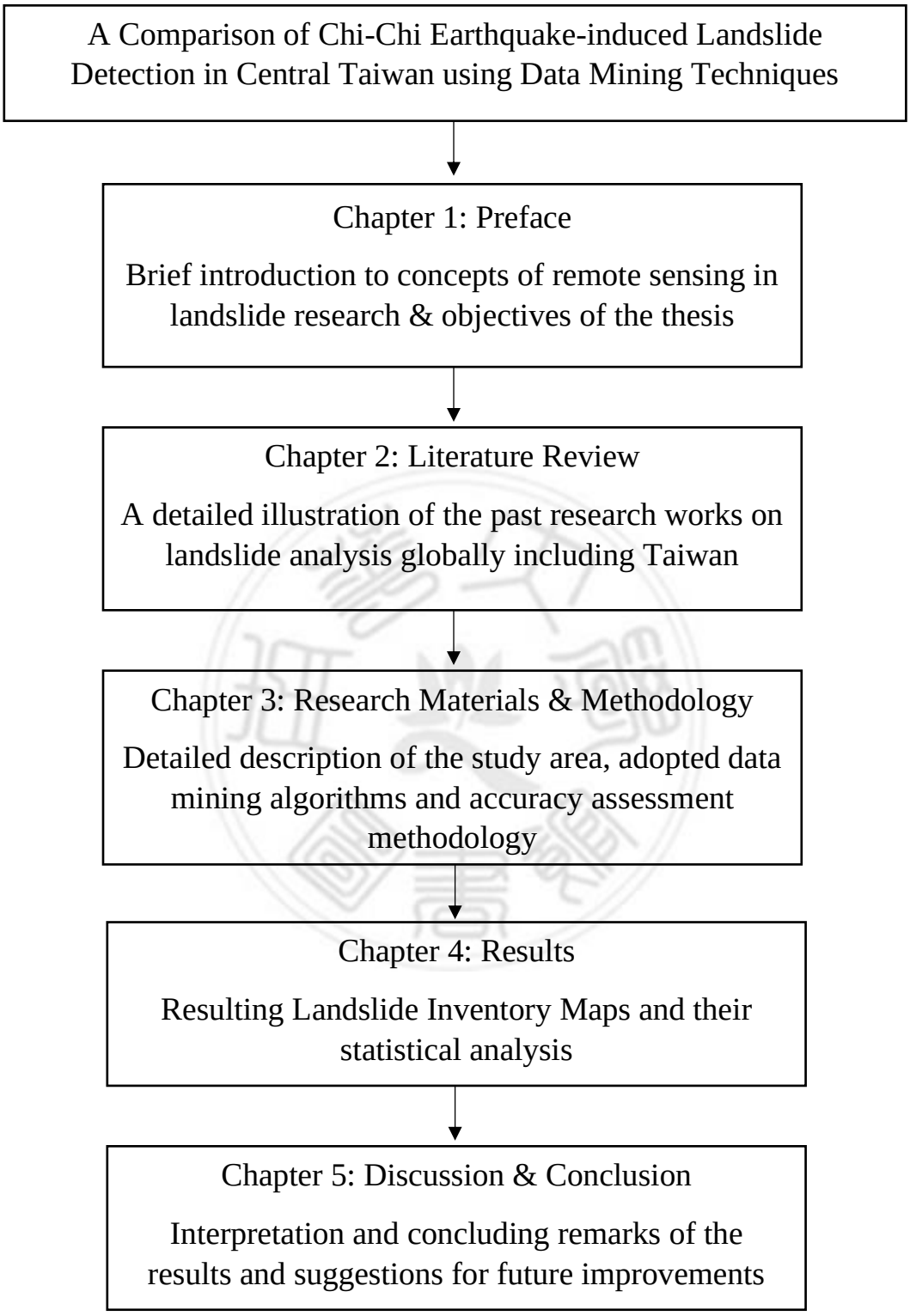


Figure 1.1: Research Architecture

Chapter 2 Literature Review

Landslide studies typically include three core aspects where detection is the foundational element on which other aspects such as prediction and warning system maximize. Landslide detection studies generally consist of mapping moderately or severely affected regions from landslides and their triggering characteristics. Landslide detection mostly includes three elements where a detailed landslide inventory map is generated: (1) a suitable feature analysis that analyses each factor associated with landslides, (2) image analysis in which an image is analyzed based on single-pixel or in coalition with its neighboring pixel and finally, (3) a technique (generally data mining) that integrates the analyzed features and image to establish a decision on whether a region is severely affected by landslides or not. However, learning about the suitability of these techniques in previous studies is vital for their applications on upcoming landslides over new terrains that were not previously studied or affected by landslides.

This chapter illustrates a detailed overview of the past landslide studies that analyzed and compared various features, image analysis techniques as well as some of the widely popular data mining techniques. A detailed survey by **Ma et. al. (2020)** includes discussion on each of the core elements associated with landslide prevention right from the beginning stages of landslide detection to the final stages of developing an early warning system. The chapter also includes studies on some other algorithms that were applied in various fields but effectively integrated remote sensing data and provides a critical analysis of their works. The review also includes some literature from Taiwan as well as some of the popular works on Mt Jou-Jou following the Chi-Chi (921) earthquake.

2.1 Feature Analysis for Landslide Detection and Susceptibility Mapping

Feature analysis involves a comparison of various factors, their selection, and fusion techniques in addition to their sensitivity to data mining techniques. Landslide maps generated should depict the interrelation between landslides and their causative factors (Pradhan et al., 2018). Hence, several studies in the past have evaluated and compared various geological and morphological information derived from remote sensing data and investigated their impacts on the predictive capabilities of some of the well-favored data mining algorithms. Features related to landslides consist of topographic, geologic, and rainfall data. **Wang (2020)** implemented these features and established three geodatabases based on recent, relict, and joint (both recent & relict) landslides and investigated five supervised machine learning algorithms on these databases for landslide identification. Their study revealed that the "Convolution Neural Networks" (CNN) model achieved the highest landslide detection accuracy of 92.5% on the recent landslide database compared to other algorithms for its powerful abilities to preprocess multi-dimensional data and feature extraction. The configuration for the most accurate CNN model was 11 layers (i.e., four convolutional, four max-pooling, and three fully connected layers) and was termed as deeper-CNN (DCNN-11). The paper concluded that all the models showed the highest accuracy on the recent landslide database among all three databases and features like Slope Gradient, Aspect, Curvature, and Topographic Wetness Index (TWI) were the prominent factors.

Feature selection plays a pivotal role in the landslide detection process and heavily influences its accuracy. **Chen (2014)** emphasized feature selection and evaluated features computed from LiDAR-derived DTM. This study proposed a framework that combines LiDAR derivatives, a method for feature selection, and the Random Forest algorithm. The generated landslide inventory map was processed

to make finer boundaries between landslide and non-landslide regions by a technique known as “Canny Operator”. The quantitative analysis of the landslide map revealed that the feature selection positively influenced the overall accuracy by 0.44% approximately. The feature set was reduced remarkably by 74% which significantly reduced the computation time and complexity of the model. The highest accuracy achieved in this study was 78.24% which is notably lower than expected.

However, possible combinations of dominant factors should also be investigated for their influence on the data mining models. In this context, **Pradhan & Mezaal (2018)** suggested the evaluation of feature selection techniques due to irregularities in results obtained from different selection techniques. The study assessed six feature selection methods and compared their results using SVM and RF. The features were the “Digital Surface Model” (DSM) and “Digital Elevation Model” (DEM) derived from LiDAR. The feature selection methods adopted for their study were “Ant Colony Optimization” (ACO), “Gain Ratio” (GR), “Particle Swarm Optimization” (PSO), “Genetic Algorithm” (GA), “Correlation-based Feature Selection” (CFS) and a Random Forest-based technique also known as embedded method (R_F). Both the algorithms i.e., SVM and RF yielded the highest detection accuracy when based on optimal features from CFS, ACO, and R_F. However, all the six feature selection methods chose identical features but provided varied combinations (i.e., ranks) and eventually yielded dissimilar landslide accuracy. On this basis, the paper briefly touches on the importance of investigating feature combinations for landslide detection. This research was conducted by subdividing the study area into two zones designated as test site-1 and test site-2. Among the two algorithms, the RF model achieved the overall highest accuracy of 88.68% (86.82%) on the optimal combination of features (all features) whereas SVM obtained 86% (84%) on site-1 and site-2, respectively. The study discovered

that the sensitivity of the SVM model to feature selection was much higher than the RF model.

Another study from **Tran et. al. (2019)** synthesized the feature selection and their effects on a predefined number of clusters for unsupervised learning. The study compared two unsupervised algorithms namely K-means and GMM for landslide identification and utilized feature extractors from LiDAR-derived DEMs. The study generated a total of four landslide maps for each topographic feature by initially defining the clusters at 2, 3, 4, and 5. Four topographic features such as Roughness, Slope, Local Topographic Range, and Variability were evaluated. Hence, a total of 16 landslide maps were generated and comprehensively compared. The comparison revealed that both algorithms achieved the highest overall accuracy of about 87%. Quantitatively, the GMM model achieved the highest accuracy of 87.09% on only roughness factor when the number of predefined clusters was 4 whereas K-means achieved 87.19% which was also on roughness but only required 2 clusters centroids.

Nonetheless, only evaluation on feature selection and training the models based on optimal features does not describe the full picture on feature analysis for landslide identification. Hence, a study from **Ghorbanzadeh (2019)** investigated the impact of layer stacking on several deep learning and machine learning models for landslide detection. This study fused spectral data from the "Rapid Eye" satellite and topographic information extracted from the "Japanese Aerospace Exploration Agency's" JAXA ALOS sensor from 2016. The study discovered that topographic features slightly decreased the landslide detection when compared to only spectral features (R, G, B, NIR, and NDVI). Although, the topographic features were helpful for the classification of settlement and landslide areas. Here, only the slope layer was beneficial because landslides generally occur on steeper slopes, unlike settlement areas. The study indicated that simply adjoining two or more suitable

features together side by side in a database will not always yield improvements. Accordingly, the study from **Sameen & Pradhan (2019)** analyzed two feature adjoining techniques i.e., layer stacking and feature-level fusion, and evaluated their influence on popular deep learning techniques such as "Residual Neural Network" (RNN) and CNN. The feature fusion techniques were conducted by combining topographic information with spectral bands. The research concluded that the feature-level fusion enhanced the model's accuracy while the network depth, architecture, and parameters remained unchanged. Interestingly, the layer stacking degraded the accuracy of all models regardless of their architecture and depth when compared to models that were only based on spectral information. The paper found that residual networks showed better performance only on the spectral dataset, whereas, CNN yielded higher performance on the topographic dataset only.

These studies indicate that the architecture of deep learning models should be designed according to the nature of the input dataset. The studies also highlight that the superiority of deep learning models over other machine learning techniques is quite limited to their depth and architecture.

Landslide Susceptibility Mapping (LSM) means locating regions that are vulnerable to landslides in the future. However, susceptibility mapping depends on the results conducted through previous landslide identification studies. The landslide susceptibility accompanies a straightforward principle: "The past and the present are the keys to the future" (Ma et al., 2020). Thus, previous studies on the area can provide ample information on major triggering factors, and climatic condition which promotes the occurrence of landslides. One such study from **Nhu (2020)** first located 152 landslides based on InSAR, Google Earth (GE), and field surveys for training three machine learning models to generate Landslide Susceptibility Maps i.e., Logistic Model Tree (LMT), Random Forest (RF), and Logistic Regression (LR).

The validation of the detected landslides was conducted on 20% of detected locations using a handheld Global Positioning System (GPS). Hence, the models were trained on 80% (122 landslides) and validated on the 20% locations (30 landslides). The models were then employed to generate susceptibility maps using 17 features. In this study, LMT outperformed other algorithms and achieved the highest rank based on the adopted evaluation metrics.

However, the previous study highlights little about the major contributing factors and only gave a general idea of the vulnerable sites which in turn could hinder the process of pinpointing the exact reasons behind vulnerability and forming mitigation strategies. Consequently, **Sahin (2020)** generated LSM using 15 causative factors and 105 landslide sites, from which 70% of the sites were used for training and the rest 30% for testing. This study built three predictive models based on "Gradient Boosting Machines" (GBM), XGBoost, and RF. The study adopted a symmetrical uncertainty measure to find the most influential factor and utilized these features to configure landslide predictive models. The most critical factors for landslides were Slope, Elevation, Topographic Wetness Index (TWI), and Sediment Transport Index (STI). Based on these, the XGBoost model achieved the highest accuracy with a kappa value of 0.9121, whereas the RF model yielded 0.8762 and the lowest value of 0.8283 by the GBM model. All the proposed algorithms in this study provided reasonable performance and declared the robustness of tree-based algorithms in landslide susceptibility. However, a study from **Pradhan & Kim (2020)** demonstrated otherwise. This research compared ensembled algorithms like RF, and XGBoost with a deep learning model i.e., Deep Neural Network (DNN) for developing a susceptibility map for shallow landslides induced by rainfall. The landslide inventory was acquired from historical sources, aerial images, and field surveys contained 748 sites for training and 219 sites for testing. The DNN model

achieved the highest overall testing accuracy of 83.71% whereas XGBoost achieved 74.73% and the RF model yielded the lowest accuracy of 68.19%. The most important factor in this study was found to be the proximity to drainage which is significantly important during higher precipitation to channel excess water accumulate over terrains. The slope factors were equally important and ranked second as the most influential cause. However, as stated earlier the performance of deep learning models strictly based on their depth and architecture and may not outperform other statistical algorithms with every single configuration (Sameen et al., 2019; Ghorbanzadeh et al., 2019)

Decision tree algorithms have been widely adopted for their simplicity and non-complex tree structure and have largely contributed to landslide susceptibility. **Alkhasawneh (2014)** proposed to compare four distinct Decision Tree models (CHAID, Exhaustive CHAID, QUEST, and CRT) by adopting the significance of 21 causative factors on landslides for susceptibility mapping. This study discovered five important factors i.e., Slope Angle, Distance from Drainage, Surface Area, Slope Aspect, and Cross Curvature that can contribute to future landslides. The highest accuracy achieved was 82% by exhaustive CHAID. Similarly, **Park (2018)** developed a relation between landslides and their controlling factors using decision tree models. However, a CRT-based decision tree was omitted from this study for susceptibility. Here, the prediction model was based on 20 factors and 548 landslide sites from which half were utilized for modeling and another half for verification. In this study, the slope was found to be the most important factor for landslides. Although, the highest accuracy was achieved by CHAID of 87.1%, while, exhaustive CHAID nearly followed suit by achieving 86.9%. The QUEST algorithm achieved an accuracy of 82.8% which was higher than the previous study by Alkhasawneh et. al. (2014), which only accomplished 74% indicating that QUEST

and CRT-based decision trees can be omitted from future landslide susceptibility studies.

These research works indicate the importance of accurately detecting historical landslides which in turn could provide information on endangered sites for future disaster that would contribute to diminishing the damages and protect ecological habitats including human settlement.

2.2 Image Analysis in Landslide Identification

Image analysis is one of the core elements in the development of a landslide detection framework. There are two types of image analysis techniques that are generally employed with remote sensing: "Object-based Image Analysis" (OBIA) and "Pixel-based Image Analysis" (PBIA). PBIA only favors one single pixel as its core analysis approach whereas OBIA favors contextual analysis that considers neighboring pixels along with its core pixel. Most of the studies in landslide detection favored OBIA for its superiority in providing more accurate landslide maps. However, its accuracy heavily relies on object homogeneity and parameter optimization. Hence, one should be careful in selecting and optimizing the parameters of an object (segment). One of the previously discussed studies by **Pradhan & Mezaal (2018)** adopted fuzzy-based segmentation (FbSP-optimizer) to optimize object parameters and obtained initial training sites for supervised algorithms. The Object-based analysis can be conducted on multiple scales. The study from **Tavakkoli et. al. (2019)** compared the conventional OBIA with the multi-scale segmentation combined through "Dempster-Shafer Theory" (DST) for classification results. This study ensembled three machine learning algorithms i.e., "Multilayer Perceptron Neural Network" (MLP-NN), Logistic Regression (LR), and RF as base classifiers instead of using them individually. Images from PlanetScope optical satellite and DEM were adopted for this research. The integration of multi-

scale segmentation by DST significantly improved accuracy across all algorithms. The accuracy ranged from 83.3% to 87.2% when only objects with optimal parameters were used. However, after integrating the DST to combine multi-scale results remarkably improved the accuracy up to 90%. The quantitative assessment also revealed that the accuracy of stacked algorithms achieved the best accuracy among all scales when combined with DST.

Object-based image segmentation can be applied to various stages of the landslide detection jointly with the Pixel-based approach. **Danneels (2007)** first suggested obtaining the pixel-based classified image from the maximum likelihood algorithm and then the image segmentation was conducted on it through double thresholding technique in conjunction with histogram-based thresholding. The results were compared with classification from ANN. The input feature used for this research was raw spectral bands and NDVI separately from the ASTER satellite. The classification based on NDVI provided the best accuracy. The paper concluded that the ANN classification method was slightly better than the proposed likelihood-based algorithm.

Some of the studies established a detailed comparison of PBIA and OBIA on landslide detection accuracy. A study from **Li (2015)** compared these image analysis techniques with two machine learning algorithms and their sensitivity to feature selection. The data was LiDAR-derived DTM and algorithms compared were SVM and RF. In this study, image segmentation centered on the automated selection of object features instead of object parameters. The object features were derived using LiDAR DTM and the feature selection drastically declined the number of object features and vaguely improved classification accuracy. Higher sensitivity was observed in OBIA towards feature selection than PBIA. The best landslide inventory map generated from this study achieved the highest accuracy of 89%.

Keyport (2018) also recommended a comparison based on unsupervised classification. The pixel-based classification was based on 11 trials whereas object-based classification included 2-step K-means clustering. The study used orthophotos and DEM data for comparison. The pixel-based classification resulted in 11 output images, with clusters ranging from 4 to 14. The best object-based image approach was determined through trial and error. For this, only the orthophotos with spectral bands were implemented. The elimination of false-positive was conducted by prior knowledge on the region by fieldwork and spatial characteristics of the objects. The overall accuracy of OBIA based on unsupervised classification was the highest 96.5% whereas PBIA achieved 94.3% indicating the superiority of OBIA over PBIA.

As previous studies indicated, the Object-based method provides superior quality landslide maps when compared to the Pixel-based approach. However, it comes at a cost of computational complexity and rigorous parameter optimization for each object. This results in the process being time-consuming when a quick assessment is required. Occasionally, the local authorities require instant results with an overall idea of severely impacted regions. Some research works favored the pixel-based approach over the object one due to its rapid production of landslide maps. A study from **Lei et. al. (2018)** accentuated this and proposed a fuzzy-based change detection technique i.e., "Unsupervised Change Detection using Fast fuzzy c-means Clustering" (CDFFCM). The proposed technique involves two steps. The first step includes the execution of "Gaussian Pyramid-based Fast Fuzzy c-means" (FCM) to acquire the landslide region. Secondly, an image difference is based on structure information to obtain landslide regions more accurately. This method was compared with the likes of other unsupervised approaches such as "Edge-based Level-set" (ELSE), "Region-based Level-set" (RLSE), and "Change Detection-based Markov

Random Field" (CDMRF). The fuzzy-based techniques achieved the highest accuracy when compared to other aforementioned methods with fewer parameters, and shorter runtime.

An additional study from **Ramos-Bernal et. al. (2018)** integrated "Change Vector Analysis" (CVA), "Chi-Square Transformation", and "Linear Regression" (LR) along with the principal components from NDVI to acquire the differenced image. The thresholding was conducted through two histogram-based methods: the statistical parameters and the secant method. The proposed methodology was purely automated and required limited human interventions. The paper obtained the highest accuracy of 84.81% when NDVI was implemented with LR for landslide detection and thresholding by secant.

However, many pieces of research adopted multi-temporal images that require tedious data collection and preprocessing which limits their usefulness during real-time emergencies. A study from **Zhai (2020)** focused on developing an "Unsupervised Single Image-based Landslide Detection" (USILD) method for rapid and automated detection of landslides which can provide real-time updates to landslide responses. The proposed method takes advantage of the visual saliency and reflectance characteristics of landslides. The final risk map was refined using morphological operators. The experimental results reported that the adopted algorithm achieved the highest accuracy of 93.72% when compared to some of the widely used supervised and unsupervised approaches. The USILD was compared with the likes of "Image Difference-based Change Detection" (CDD), "Change Detection using Ratio" (CDR), K-means, and SVM. The proposed method requires no training samples and comes with a lower computational cost.

Taiwan is one of those countries that is regularly distressed by landslides due to several factors. Such geohazard in Taiwan mainly occurs due to geologic

excitation during earthquakes or volcanic events, or sometimes induced due to heavy torrential rainfalls during typhoons that suddenly increases the accumulated water over the top surface and groundwater to a critical level that ultimately destabilizes the slopes. Hence, the majority of research works in Taiwan solely focused on locating landslide terrains and developing early warning systems based on their contributing factors using a pixel-based approach for its quick and reduced complexity. A study from **Tsai (2010)** detected landslides induced by extreme rainfall during Typhoon Morakot on 8th August 2009. These landslides resulted in sediment flow with enormous volume that instigated devastating damages to property and infrastructure causing heavy human casualties in southern Taiwan. A systematic approach utilizing multi-temporal satellite imagery with spectral indices filtering and geo-spatial analysis was proposed for accurate post-disaster assessment. Specifically, landslides were located with "NDVI filtering", "Change Vector Analysis" (CVA), and "Post-analysis Editing". A spatial analysis was conducted to develop a relationship between the identified landslides and topographic factors. This study identified a total of 9333 landslides constituting an area of 22,590 ha. Larger sections of the detected landslides were less than 10 ha. A total of 45% of the detected landslides were larger than 10 ha and the spatial analysis discovered the elevation range between 500 m to 2000 m and slope gradient was within 20° and 40°. Additionally, a specific landslide that devastated a river-side village was also examined. The average debris flow of this landslide was estimated to be higher than 2.6 million m^3 with an average depth of 40m.

Another severe landslide that devastated Taiwan was induced due to an earthquake on 21st September 1999 (921 or Chi-Chi). The landslide imposed severe destructions to natural habitat, infrastructure, and caused a higher death toll. This natural hazard particularly affected the central part of Taiwan. Landslide assessment

on Shei Pa National Park which was also one of the affected regions and a victim to strong ground movement from the Chi-Chi earthquake was in the progress of recovering. However, the restoration process was severely obstructed when Typhoon Toraji hit central Taiwan. Research from **Wan et. al. (2010)** provided a detailed landslide assessment in this area. The study utilized DEMs, and SPOT images to investigate several conditional factors and proposed a data mining technique i.e., "Discrete Rough Set" (DRS) to be compared with the C4.5 decision tree model. The study found the DRS method to be a superior classifier than a conventional C4.5 decision tree. The core attributes for landslide occurrence were Vegetation index (VI) and NDVI during the observation period indicating vegetation condition as the major governing factor for landslides in the Shei-Pa area.

The mountainous region of Chiufenershan in Central Taiwan was also severely damaged by the Chi-Chi earthquake and induced several largescale landslides. A detailed landslide survey of the area was adopted by **Lin et. al. (2008)** revealed that the estimated landslide area was 215.68 ha soon after the earthquake. The study was conducted through multi-temporal SPOT satellite images which were dated between 1999 to 2006 coupled with "Self-Organizing Map" (SOM) a type of unsupervised neural network, terrain analysis, and a "Universal Soil Loss Equation" (USLE) to indicate landslide patches. This study also proposed a new spectral index i.e., Total Brightness Index (TBI) as an alternative for NDVI. Till 2006, the patch area reduced to 113.96 ha indicating a 47.44% of the total landslides have recovered. According to terrain analysis of the denudation sites, debris volume was estimated to be 31,896,667 m^3 and 39,537,067 m^3 at collapsed and deposited sites, respectively. The erosion depth annually just after the earthquake was 22.07 mm which was about 3.59 times pre-earthquake. As a result of landslide restoration, the annual erosion was declined to 13.54 mm of about 2.21 times pre-earthquake. The

obtained result from this study indicated the efficacy of nature itself to handle ecological restoration without any human interventions.

Mt Jou-Jou another gravely damaged mountainous terrain from the Chi-Chi earthquake was extensively monitored for its vegetation restoration. Several studies evaluated and monitored vegetation recovery rate post-earthquake on a short-term and long-term basis. For this, the local government authorities reserved a small portion of the mountain as a reservation park following the earthquake. A series of literature is available on the restoration process of Mt Jou-Jou. **Lin et. al. (2004)** developed a "Vegetation Restoration Rate" (C) to monitor vegetation succession one year after the earthquake. The landslide characteristics were studied through integrating NDVI with multi-temporal SPOT images that were coupled with GIS, aerial images, and in-situ field investigations. The estimated restored region to be 73.35% of the total area with an annual recovery rate of 47.1% indicating about half of the landslide sites have been restored after one year. The research also designated poor to very poorly recovered regions to be mountain ridges, scoured slope bases, and acidic sulfate soil areas due to the effects from soil moisture and SO_4^{2-}. The distribution of landslides covered a total of 908.96 ha over the entire mountain range was illustrated clearly in this research.

Another study from **Lin (2006)** assessed vegetation and computed soil erosion after six years following the Chi-Chi earthquake. This study also utilized NDVI derived from multi-temporal SPOT satellite imagery. After six years since the earthquake, the denudation area reduced to 143.22 ha indicating that the majority of the landslides have been recovered. The vegetation percentage over landslide area reached 89.69% showing a significant vegetation improvement over the previous study. The soil erosion depth during the initial stages of the earthquake was 6.04 mm, about 2.26 times pre-quake. However, the field surveys showed the formation

of vegetation buffer strips by several pioneer plant species to mitigate the impacts of debris flow and soil erosion. The annual soil erosion depth reduced by 47.48% about 1.18 times pre-quake and was estimated to be 3.16 mm on average.

However, previous studies were disputed by the research from **Yang (2017)** based on limited satellite image usage that could severely delude the statistical regression model and no consideration of seasonal variation on vegetation recovery. As the projected recovery rate from Lin et. al. (2004) and Lin et al. (2006) were based on short-term observation these research works could not provide sufficient information on the long-term trajectory. Hence, a total of 14 SPOT satellite images from 1999 through 2011 were utilized to derive a long-term trajectory for vegetation recovery over Mt Jou-Jou. The study comprehensively integrated multi-sensor satellite images from SPOT 1, 2, and 4. During the course of 12-year vegetation succession, the average NDVI for total area and the landslide areas rose from 0.278 to 0.431 and -0.044 to 0.367, respectively. Two separate "Vegetation Recovery Rate" (VRR) models were proposed: one being the original VRR and another was a modified-VRR with seasonal adjustment. The modified-VRR approached a value of 81.5% and 81.3% for total and landslide area, respectively. The regression model for modified-VRR on total area achieved an R^2 value of 0.915 which was a significant increase over the traditional VRR having an R^2 value of 0.584. As for the landslide area, the R^2 value showed an improvement from 0.883 to 0.916 with the seasonal adjustment that recommends its significance in vegetation recovery assessment.

The cited literature in this section emphasized the contribution of image analysis techniques in the field of landslide detection. However, both approaches yielded better accuracy than the other in some of the studies their applicability is solely a case-by-case phenomenon. OBIA requires rigorous parameter optimization

and heavily depends on similarity within the object which is time-consuming, and complex. PBIA is fast and efficient which requires no prior optimization and can configure quick, efficient, and automated landslide detection systems. However, the accuracy based on it highly depends on several factors such as orthorectification, and co-registration, it is also very sensitive to noise and outliers in the pixel values. Hence, with proper necessary preprocessing methods when an emergency quick response system post geohazard is to be developed most studies adopted PBIA whereas a detailed analysis on triggering factors was based after segmenting (OBIA) the initial raw images.

The development of landslide detection along with its contribution towards landslide analysis has been thoroughly discussed with the necessary background and relevant literature survey. Each of the core elements such as appropriate feature analysis, image analysis, and data mining techniques have significantly contributed to providing robust and efficient geohazard monitoring systems. The landslide inventory has also provided ample information on historical landslides that could further the advancements in landslide prediction systems. The reviewed literature in this chapter indicated that the slope was the most prominent factor for inducing landslides on mountainous terrains with steeper gradients. Other factors instigating instability on slopes were reduced vegetation presence and improper drainage. A proper drainage system on slopes is crucial to prevent rainfall-induced landslides. The study from Pradhan et. al. (2020) demonstrated it by developing a susceptibility model for rainfall-induced landslides. The mitigation strategies for these landslides may include the construction of adequate drainage channels to drain excess water accumulated on slopes. However, slope destabilization does not only occur during rainfalls but also through landscape excitation during the earthquake or volcanic outburst. Hence, proper precautions shall be taken to stabilize these slopes during

such calamities. This can be accomplished through the construction of retaining walls, or earth anchors. Although, these structures can be costly if the slope gradient is higher or the slope is near a man-made structure such as roads, highways, bridges, etc. Hence, establishing an adequate vegetation buffer strip to stabilize these slopes would be a cost-effective and eco-friendly solution that provides dual benefits of preventing landslides from earth excitation or torrential rainfalls. On this basis, the literature from Taiwan primarily focused on monitoring vegetation restoration following some of the devastating calamities of the decade this country has ever faced. Each of these studies solely focused on providing a fair share of information on vegetation conditions from pre-slide and post-slide dated satellite imagery which is an economical source of data. Usage of multi-temporal remote sensing images significantly reduced the field surveys and provided sufficient knowledge within a limited timeframe which resulted in quicker response to the natural calamities. These studies aided local government authorities and ecological engineers alike to improve and develop genuine mitigation strategies for landslide restoration based on vegetation. Studies on Mt Jou-Jou, Taiwan mainly focused on implementing pixel-based change detection using image differencing to locate landslide areas illustrating potential integration and assessment on data mining algorithms for this region.

2.3 Literature on Other Data Mining Algorithms

Some of the algorithms adopted for this research such as Minibatch K-means, BIRCH, Extra Trees (ET) are not widely adopted in landslide studies. In this section, we reviewed a handful of research works on their applications with remote sensing or other general purposes to gain a sense of their applicability. Moreover, additional literature on GMM is also reviewed as its adoption was quite limited in landslide studies and to draw out its theoretical background for this research which was not properly discussed in the previous sections.

Ari & Aksoy (2010) proposed a "Particle Swarm Optimization" (PSO) for parametrization in GMM algorithm which conventionally uses "Expectation-Maximization" (EM) as the standard algorithm for parameter estimation. This study highlighted the problems associated with EM getting trapped in local-maxima and suggested the PSO algorithm for faster convergence. The quantitative results showed that PSO-based GMM achieved accuracy in 60 iterations closer to its EM-based counterpart which took 500 iterations to achieve the best results.

Li & Fowler (2013) investigated the feasibility of GMM and "Markov Random Fields" (MRF) for classifying hyperspectral images. Due to large-scale parameter space in the GMM algorithm, its adoption has been significantly reduced with hyperspectral image data. This issue was countered by dimensionally reducing the data by two techniques with the preservation of multimodal structures. The input data for the GMM model consisted of spatial information from MRF. The experimental results demonstrated superior results from the proposed methodology and yielded the highest accuracy of 94.96%.

Hsu et. al. (2017) proposed a CNN-based joint clustering technique for partitioning large-scale datasets. The technique proposed to utilize a pre-trained CNN model to extract the initial cluster. The Minibatch K-means algorithm was deployed to assign a cluster to each subset (mini-batch) of the image where the CNN model subsequently optimizes all parameters iteratively through "Stochastic Gradient Descent". The proposed algorithm exceeds some of the widely adopted clustering techniques for large image-scale datasets containing millions of images.

Tan (2019) proposed an "Object-based approach for Automatic Change Detection using Multiple Classifiers and Multi-scale Uncertainty Analysis" (OB-MMUA). The study proposed to fuse SVM, K-Nearest Neighbor, and Extra Trees (ET) as the base classifiers using "Dempster Shafer theory" (DST). The input data

contained features from “Gray-Level Co-occurrence Matrix” (GLCM), morphology, and “Gabor Filter Texture” along with spectral information. The initial training sample was acquired using CVA-based uncertainty analysis. The optimal feature vectors were determined using the RF model. Hence, the object-based DST fusion and uncertainty analysis were integrated to classify the differenced image. The quantitative evaluation established the superiority of the proposed methodology for change detection which achieved the highest overall accuracy of 96.98% and 93.1% on ZY-3, and GF-2 satellite datasets, respectively.

Hardy (2019) utilized the Extra Trees Classifier for mapping mosquito breeding habitats by automatically detecting the open and vegetated water bodies. The study employed Sentinel-1 satellite imagery in Western Zambia and integrated radar satellite with machine learning as a potential element for disease elimination campaigns. This research emphasized the importance of detecting vegetated water bodies that provide an adequate environment and refuge to mosquito species where aquatic life is limited. The proposed methodology achieved an overall accuracy of 92% indicating its significance towards disease prevention programs in Africa.

Sathiaraj (2019) predicted climate types using clustering algorithms in the United States. The study adopted K-means, BIRCH, and DBSCAN for comparison and included daily climate data measurement for temperature over the period between 1946 to 2015. The K-means and BIRCH algorithm provided high-quality clustering solutions whereas DBSCAN failed in effective clustering partly due to its weakness in scaling high-dimensional datasets.

Reddy et. al. (2020) adopted the Minibatch K-means algorithm to classify hyperspectral remote sensing images and compared them with K-means, and “Density-based Spatial Clustering of Applications with Noise” (DBSCAN). In this

study, the K-means algorithm appeared to be superior and better for hyperspectral imaging data.

Ren (2020) focused on comparing strategies to reduce the dimension of hyperspectral images and their subsequent partition. For this, the study adopted "Principal Component Analysis" (PCA), and the "original Relief-F" for dimensionality reduction. Furthermore, the dimensionally reduced image was partitioned using K-means and BIRCH which were subsequently compared with the adopted "Partitioned Relief-F" method. The research was conducted over three different datasets and the proposed technique achieved superior classification results.

Cintia et. al. (2020) developed a movie recommendation system based on unsupervised clustering algorithms. Nine algorithms such as "K-means", "BIRCH", "Minibatch K-means", "Mean-Shift", "Affinity Propagation", "Agglomerative Clustering", and "Spectral Clustering" were compared. The comprehensive comparison of these algorithms revealed that the BIRCH algorithm yielded the lowest mean square error and was regarded as the best method.

The cited literature and survey process indicated that statistical, and deep learning models such as SVM, RF, and CNN were widely popular and largely adopted for landslide detection. Decision trees and Sequential tree-based ensembled boosting algorithms such as XGBoost and GBM were popular among landslide susceptibility mapping. Unsupervised clustering algorithm such as K-means was widely adopted as a comparison benchmarking tool for showing the superiority of the proposed corresponding automated methodology. Other unsupervised algorithms such as BIRCH and Minibatch K-means were not at all applied on landslide studies and only integrated for other purposes with remote sensing data. These algorithms exhibited exceptional accuracy and performance proclaiming their

robustness. Similarly, the GMM algorithm also had very limited applications on landslide identification, and other studies only focused on its potential optimizations.

Chapter 3 Research Materials & Methodology

3.1 Study Area

Mt Jou-Jou is situated near the northern shore of Wu-Chi River in Nantou County, Taiwan having 4396 ha of watershed area, and the altitude ranges between 123 and 776 m (Lin et al., 2006). The geographic location of Mt Jou-Jou in Taiwan is shown in Figure 3.1. The surface on the upper layer of Mt Jou-Jou is mainly categorized by Pliocene-Pleistocene Tokazan beds. The slopes vary between 50° and 85° from which almost half of the slopes lie within 60° to 70° (Yang et al., 2017). The gravel layer in this region has very high hydraulic conductivity with a thickness of 1000 m. Sometimes, high consolidation occurs in this layer causing it to harden during dry conditions. The surface geology allows the formation of several deep gullies on Mt Jou-Jou due to its higher erosion into deep layers throughout the rainy season, as the gravel hardens during dry conditions the surface becomes rougher (Lin et al., 2004; Lin, 2006; Yang et al., 2017). Mt Jou-Jou strongly comprises of geologic Tokazan formation, which mainly composed of three layers where the upper layer is made of bulky rock gravels, the middle layer is an assembly of clayey sand with gravel interbed, and finally, the lower layer that constitutes sand and shale with a thin interbed made of gravel (Yang et al., 2017). Mt Jou-Jou experiences precipitation of 1800 mm annually with an uneven spatial and temporal distribution (Yang et al., 2017). After the Chi-Chi earthquake, a five-year plan for an in-situ rainfall precipitation monitoring program was proposed. During the rainy season (April-September) 80.2% of the rainfall occurs, while the remaining 19.8% happens throughout the drier season (October-March) (Yang et al., 2017). Mt Jou-Jou tolerated shallow failures periodically due to high and short rainfalls including a few of the ground shaking events. The vegetation in cliff areas is mainly composed of "*Formosan giantreed*", "*Arundo Formosana*". The "*Taiwan Red Pine*", "*Pinus*

Taiwanensis Hay", "*Taiwan Short Leaf Pine*" and "*Pinus Morrisonicola Hay*" are the dominant species of trees found in moderately sloped land and hillside bases (Lin et al., 2004; Lin et al., 2006). Mt Jou-Jou is also composed of other woody plant species such as "*Trema Orientalis*", "*Ficus Erecta Thunb*", "*Koelreuteria henryi Dumm*" and *"Cyclobalanopsis galuca*" (Yang et al., 2017). In response to this catastrophic disaster in Central Taiwan, the Council of Agriculture in Taiwan decided to retain an area of 1198 ha for nature conservation purposes. Under the Taiwan Cultural Heritage Protection Law, the landslide area was maintained as a reservation park and adopted for studies related to vegetation recovery assessment.

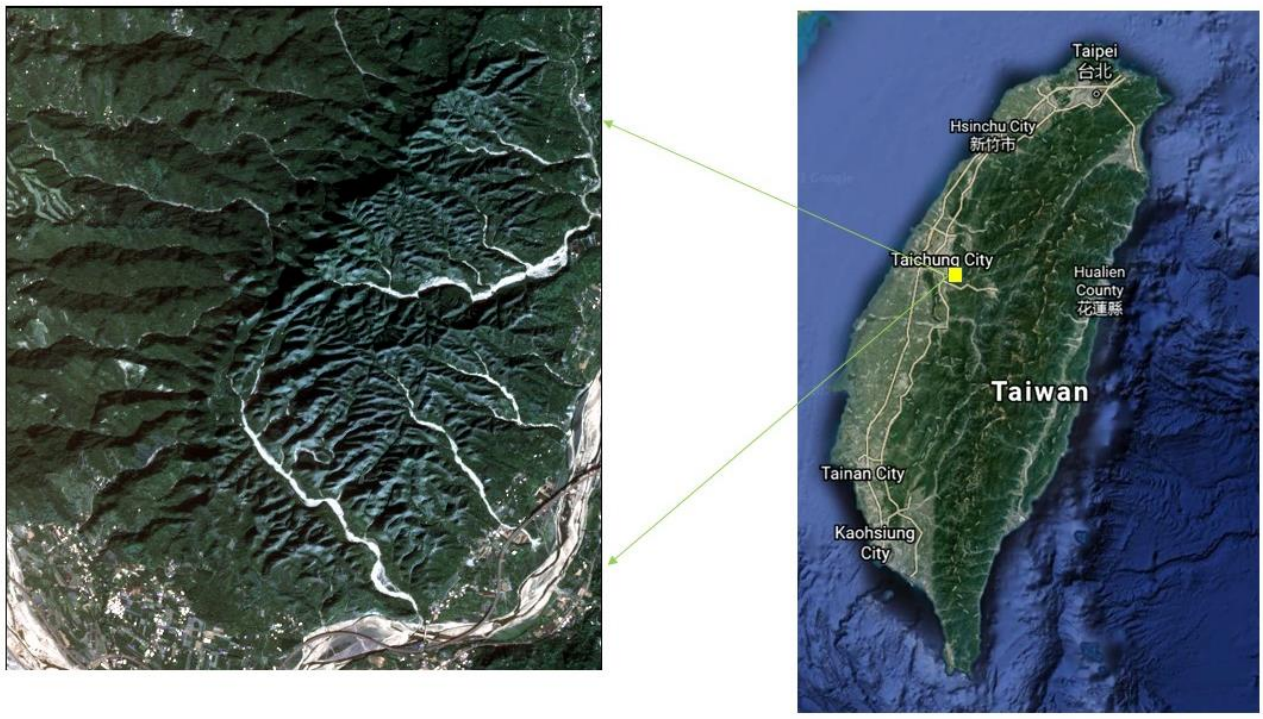


Figure 3.1: Mt Jou-Jou in Central Taiwan

3.2 Landslide Identification in Mt Jou-Jou, Central Taiwan

As discussed earlier, this research proposes a comparison of nine Landslide Inventory Maps obtained from four unsupervised and five supervised algorithms. The proposed methodology was divided into six steps:

(a) Remote sensing satellite image acquisition from pre-quake and post-quake dates along with its preprocessing.
(b) Input features such as Normalized Difference Vegetation Index (NDVI) and Total Brightness Index (TBI) were calculated, differenced, and stacked together in a same data frame from pre-processed multi-temporal images.
(c) Extraction and pre-processing of training, and testing set from satellite imagery in each category i.e., Landslide (L) and Non-Landslide (NL).
(d) Training & testing of the supervised algorithms based on hyperparameter tuning along with monitoring for overfitting and underfitting.
(e) Landslide Inventory Map generation from trained supervised models as well as unsupervised clustering algorithms.
(f) Quantitative accuracy assessment and comparison of all acquired landslide maps based on a validation set with accurate ground truth.

Subsequently, a detailed discussion on each step for the proposed methodology has been given in the following sections. A detailed workflow diagram of the adopted methodology is demonstrated in Figure 3.2.

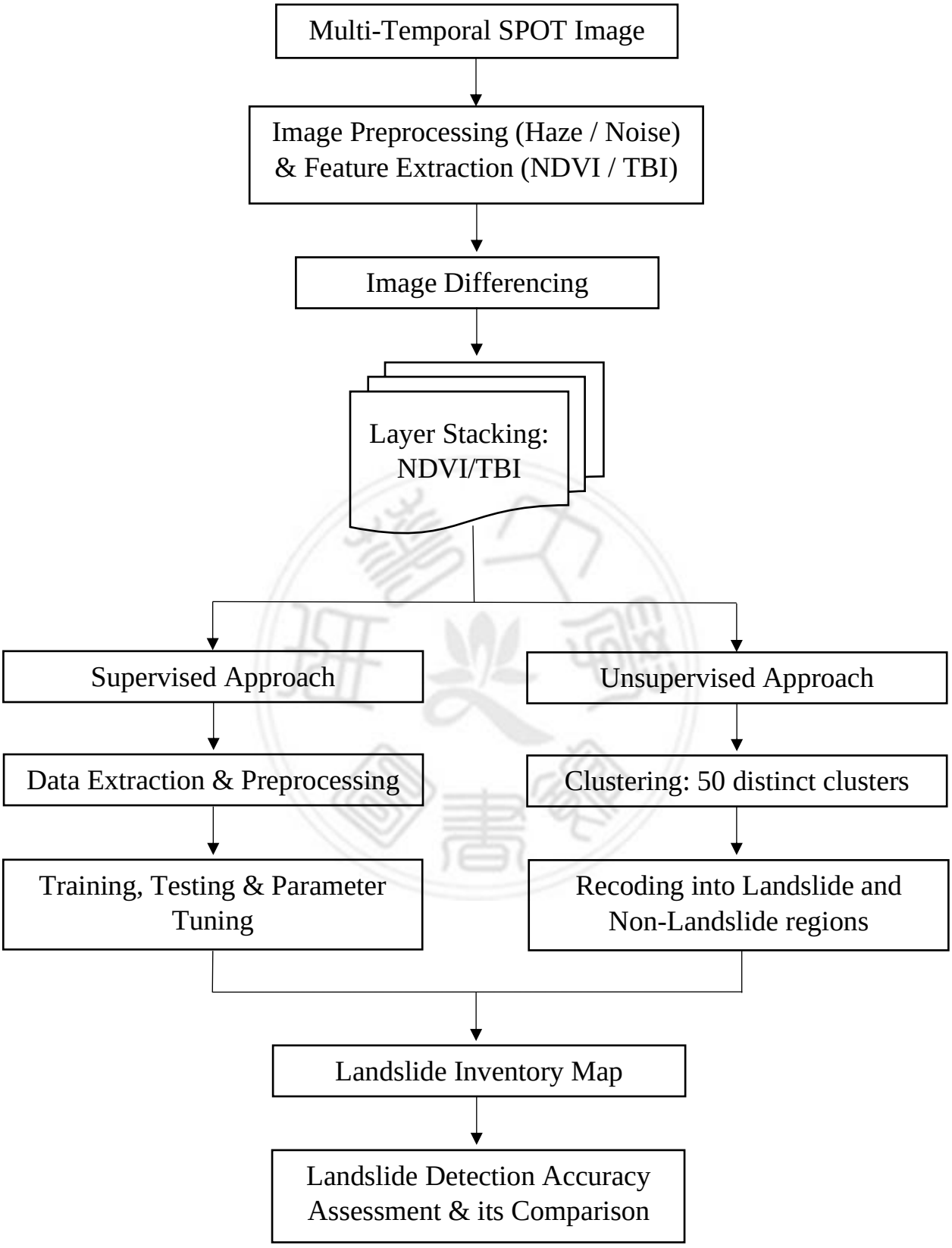


Figure 3.2: Schematic Framework of the Proposed Methodology

3.3 Remote Sensing Data and Feature Extraction

Data extracted from remotely sensed multi-temporal satellite imagery has greatly contributed to monitoring natural hazards and natural resource management. Images from the SPOT-2 satellite were acquired over the entire study region. The image acquired on 1st April 1999 represents the pre-earthquake condition, while the image dated 27th September 1999 represents immediate post-earthquake conditions, shown in Figure 3.3. The SPOT satellite imagery mainly consists of 3-bands, green (G), red (R), and near-infrared (NIR). Each pixel is sized 12.5 x 12.5 (m) and the image contains a total of 500 x 558 pixels covering the entire study area. The reflected wavebands underwent two corrections, haze correction followed by noise correction. Both corrections were implemented using high-caliber geospatial software. Due to atmospheric effects at the time of image acquisition, the image may have a limited dynamic range causing image appearance with very low contrast and high haziness. Haze correction consists of two techniques "Tasseled Cap Transformations" or "Point Spread Convolution". This technique yields components that correlate with haziness. The method removes this component and the image is transformed back to RGB space. In this study, the Point Spread Convolution technique was utilized to remove the effect of haziness on the image. Noise correction removes the portion of noise in the raster layer that preserves the finer details in the satellite image such as thin lines while removing noise along the edges and flat areas.

Two spectral indices are stacked together as input layers for these algorithms. Firstly, the Normalized Difference Vegetation Index (NDVI) was proposed to evaluate the extent and localization of vegetation in the study area. The characteristics of the chlorophyll imply that red wavelength has high absorption whereas infrared waveband has a high reflectance. This indicates that spectral

response from vegetation can be measured by reflected red over infrared wavelength (Yang et al., 2017). NDVI is one of the most favored and versatile vegetation indices for vegetation monitoring (Lin et al., 2006). The NDVI (Lin et al., 2004; Lin, 2006; Yang et al., 2017) can be calculated as:

$$NDVI = \frac{NIR-RED}{NIR+RED} \quad \ldots\ldots (1)$$

Where NIR is the reflectance of the infrared wavelength and RED is the reflectance of the red wavelength. The NDVI value ranges between -1 and 1.

A high NDVI value indicates that the area consists of very high or dense vegetation whereas a low NDVI value indicates bare soil, denudation area, or even landslide. These characteristics were utilized to investigate the drastic change in vegetation from pre-earthquake and post-earthquake images to identify the location of landslide-affected regions. NDVI was adopted as a vegetation indicator for post-disaster landslide assessment in southern Taiwan due to Typhoon Morakot (Tsai et al., 2010). Recently, NDVI derived from various remote sensing satellite images have been coupled with several topographic features extracted from DEM, rainfall data, geological data (lithology), hydrological data for landslide detection, and susceptibility mapping in the high mountainous regions (Wan et al., 2010; Bernal et al., 2018; Tavakkoli, 2019; Nhu, 2020; Sahin et al., 2020).

Another evaluating index was fused with the spectral vegetation index as input layer i.e., Total Brightness Index (TBI) (Lin et al., 2008). TBI was implemented to enhance the spectral feature of the landslide and was proposed as a substitute for near-infrared reflectance. The index is computed by the following expression:

$$TBI = \frac{G+R+NIR}{3} \quad \ldots\ldots (2)$$

In which G is the green band reflectance, R represents the reflectance in the red wavelength, whereas, NIR is the reflectance for the near-infrared band. The proposed TBI displayed strong reflectance and enhancement on the landslide region. Therefore, landslides can be extracted by utilizing the brightness values from G, R, and NIR. TBI was adopted as an evaluating factor for landslide extraction from multi-temporal SPOT images in Chiufenershan which was also severely affected by the landslides induced due to the Chi-Chi earthquake (Lin et al., 2008).

In this study, both NDVI and TBI derived from SPOT satellite images acquired from pre-earthquake and post-earthquake images were differenced for each value pixel-by-pixel by "Image Differencing" which is also known as "Change Detection using Image Differencing" (CDD) (Zhai et al., 2020). The process is simple, it includes standard subtraction of digital values of each pixel from a pair of given images and produces a third image consisting of numerical difference between pair of pixels corresponding to the same coordinates (Lin et al., 2008). Landslides are one of the land covers that changed between the multi-temporal SPOT images, which can be extracted using CDD. The differenced images of NDVI and TBI are shown in Figure 3.4. It is apparent from the differenced images that NDVI and TBI develop an inverse relationship for the landslide region. The NDVI changed negatively whereas TBI underwent a positive change for the area experiencing a drastic reduction in vegetation. If the NDVI changed positively, and TBI had a negative change then the area witnessed improved vegetation condition. If very little change was observed between NDVI and TBI values, then the region did not go any major changes in its land cover class and is assumed to have unaffected by the natural hazard.

(a) (b)

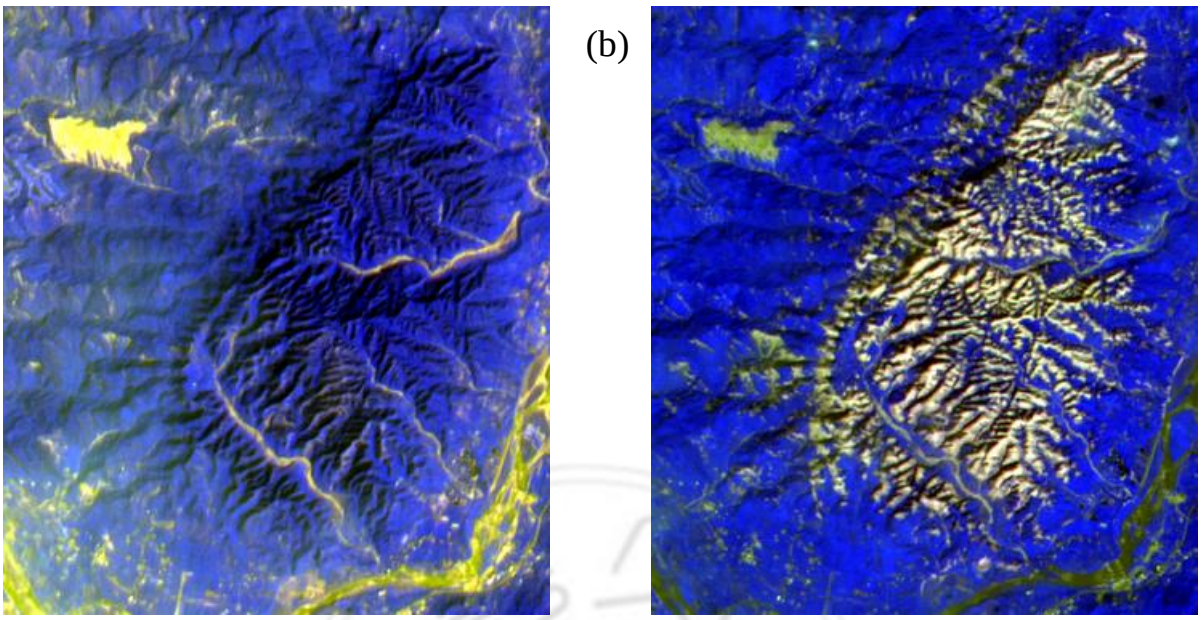


Figure 3.3: Multi-temporal SPOT Images of Mt Jou-Jou (a) 1st April 1999 (b) 27th September 1999

(a) (b)

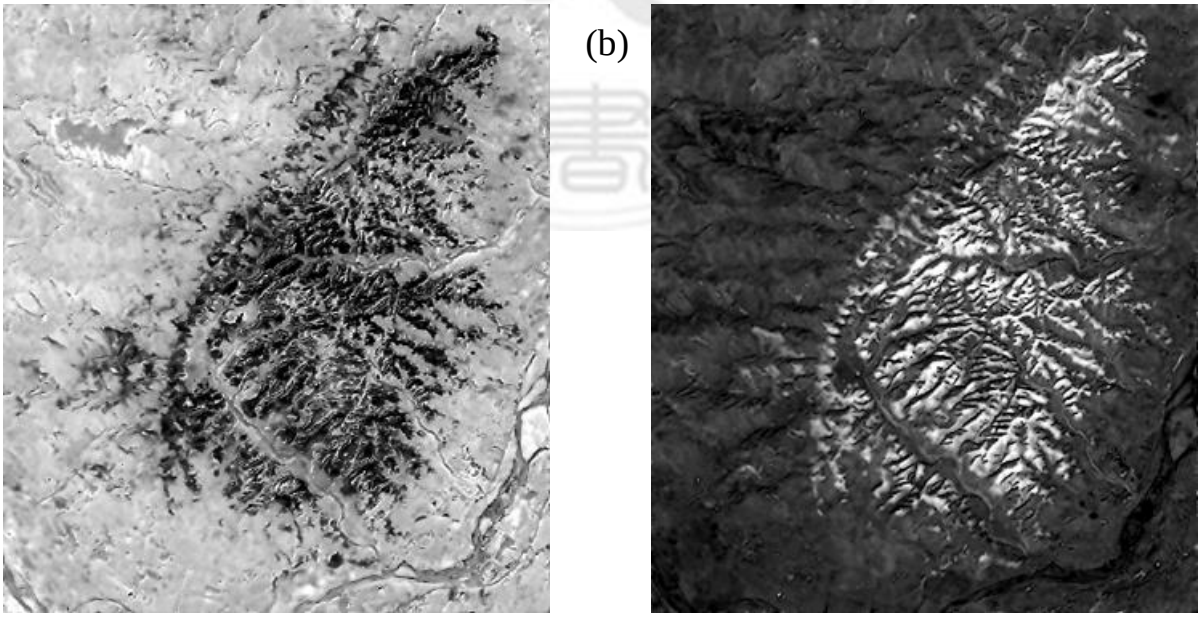


Figure 3.4: Differenced Images (a) NDVI (b) TBI

3.4 Database Extraction and Preprocessing for Landslide Detection

Establishing a robust database is critical for data mining models, that include three components: layer stacking, data extraction, and database establishment (Wang et al., 2020). In this study, layer stacking involves a simple fusion of two feature variables side-by-side together in a single database i.e., NDVI and TBI. Data extraction includes a careful selection of representative pixels for each category, this was conducted by the user's prior knowledge and cited literatures of the study area. The area of interest (AOI) feature represented by polygons was utilized to highlight and extract the coordinates of the representative pixels in each category as shown in Figure 3.5. The pixels were binary encoded into non-landslide [1,0] and [0,1] landslide region.

(a) (b)

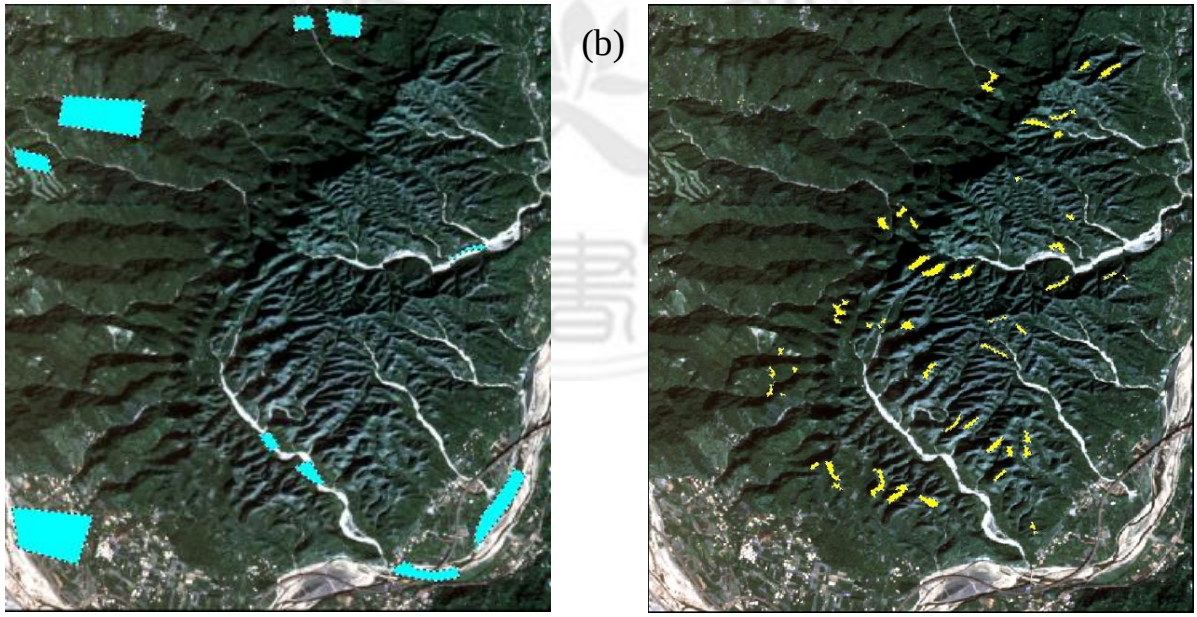


Figure 3.5: Representative Sites for Training the Supervised Learning Algorithms (a) Non-Landslides (b) Landslides

An intelligent and efficient system requires a comprehensive dataset (Alkhasawneh et al., 2014). Hence, 12316 pixels were extracted for this study, of which 3300 pixels represented landslide (L) and 9016 pixels represented non-landslide (NL). The dataset was proposed to split into two sets, a training set, and a testing set. The training set contains 25% of the whole dataset which was used to train the supervised models, whereas the rest 75% of the dataset was defined as the testing set to evaluate the classification accuracy of the trained model. The resulting dataset was highly imbalanced with landslide pixels being in the minority class, which were balanced by an over-sampling technique known as "Synthetic Minority Oversampling Technique" (SMOTE).

A frequent impediment for machine learning in Landslide Inventory Map development is the class imbalance (Ma et al., 2020). Most of the machine learning algorithms require extracting specific features related to non-landslide and landslide regions, to describe a clear classification boundary between the two categories (i.e., non-landslide and landslide). In this study, the landslide regions appear to have fewer pixels compared to the non-landslide region. Such conditional imbalance can cause the trained model to be biased towards the majority class and causing the misclassification of landslides into a non-landslide. After an investigation by Ma (2020), it was found that the RF model under-classified landslide region when data samples were highly imbalanced i.e., the dataset contained more pixels for non-landslide areas. For overcoming such problems, several techniques can be employed which are divided into 3 categories as Data-level, Algorithm-level, and Hybrid approaches (Ma et al., 2020). Data-level techniques mitigate class imbalance through diverse data resampling, algorithm-level techniques involve cost-sensitive approaches to solve class imbalance which includes modifying the algorithms themselves by the class penalty or class weights. Hybrid approaches are strategic

combinations of data level and algorithm level techniques. In a recent study, it was indicated that both instances, synthetic and original, were correctly classified by the RF classifier after preprocessing the data by adopting minority oversampling technique with iterative partitioning filter i.e., "SMOTE-IPF" (Synthetic Minority Oversampling Technique- Iterative Partitioning Filter) which is a type of data level technique (Ma et al., 2020).

For the present dataset, the minority class was landslides with 3300 pixels compared to 9016 pixels in non-landslide. Following the split, total pixels in the training set were 3079 which included, 781 samples as landslides and 2298 samples in non-landslide. The test set includes 9237 samples, in which 2519 samples were landslide and 6718 were non-landslide resulting in a highly imbalanced dataset and would likely result in supervised learning biased towards non-landslide with under-classification of landslides generating landslide map with diminished accuracy.

Hence, this study adopted a data level oversampling technique called "Synthetic Minority Oversampling Technique" (SMOTE) to eliminate the imbalance in the dataset. This approach utilizes the K-nearest neighbor to generate new instances of the minority class. It involves the selection of an instance "a" randomly in minority class while its K nearest minority class neighbors is determined. The synthetic instant is then generated by random selection from one of the K nearest neighbors "b" followed by adjoining "a" and "b" through a line segment in the feature space. The synthetic instances are synthesized by a convex combination of random instances "a" and "b". This technique was implemented using "Imbalanced-learn", an open-source sampling library for python programming language. All the parameters in this package were kept at default while oversampling the training and testing dataset.

Both input layers (NDVI and TBI) were first randomly shuffled and then oversampled followed by rescaling in the range of 0 to 1 by a technique known as "Data Normalization". It was employed using the function "Normalizer" in the "preprocessing" module of the "scikit-learn" open-source library for python. General mathematical expressions (Tan et al., 2019) of normalization for both features are given in equations (3) and (4).

$$NDVI_{norm} = \frac{NDVI - NDVI_{min}}{NDVI_{max} - NDVI_{min}} \quad \ldots\ldots (3)$$

$$TBI_{norm} = \frac{TBI - TBI_{min}}{TBI_{max} - TBI_{min}} \quad \ldots\ldots (4)$$

Where, $NDVI_{min}$ and TBI_{min} are the minimum values of NDVI and TBI, respectively and correspondingly $NDVI_{max}$ and TBI_{max} are the maximum values.

3.5 Data Mining Techniques

3.5.1 Supervised Approach

This approach requires pre-specified labeled data to build a model and develop predictive systems. The prediction accuracy for this approach is highly dependent on the parameter tuning, nature, size, and type of labeled data. Some of the algorithms in this approach are highly sensitive to noise, outliers as well as class imbalance. However, several data pre-processing techniques, as well as in-built features in some algorithms have been developed to reduce the sensitivity to these abnormalities in a given dataset. For the past few years, the supervised machine learning algorithms greatly contributed and showed promising results on remote sensing data. As per the literature review, these algorithms have been successfully combined with high-resolution spectral images derived from a variety of satellite remote sensors. Supervised learning algorithms consist of two broad categories: (1) machine learning, and (2) deep learning algorithms. Machine learning algorithms

parse the data, develops a decision boundary while training, and does prediction, whereas deep learning algorithms have brain-like logical structures in layers known as “Neural Networks” which can make much more intelligent decisions on their own. Widely used deep learning algorithms are “Artificial Neural Network” (ANN), “Convolution Neural Network” (CNN), “Deep Neural network” (DNN). The literature review indicated that recent researches mostly considered “Support Vector Machines” (SVM), and “Random Forest” (RF), which comes under the machine learning category for predictive modeling of landslide detection, as well as deep learning algorithms such as ANN, DNN and CNN have also significantly contributed to this field. All the algorithms adopted in this study are implemented through the open-source library i.e., “scikit-learn” for python programming.

3.5.1.1 Support Vector Machines (SVM)

SVM is a kernel-based non-parametric statistical algorithm that transforms the given data into a higher dimension to develop a hyperplane that consists of a non-linear decision boundary for classification. SVM has been widely applied and contributed to several landslide studies (Li et al., 2015; Pradhan et al., 2018; Ghorbanzadeh, 2019; Tan, 2019; Ma et al., 2020; Wang et al., 2020) due to its reduced dependency on a large dataset and less training time. SVM is considered to be highly adaptable and provides good performance under various settings (Wang et al., 2020). The technique by which SVM enlarges the feature space of the given data is called the “kernel trick”. Four available kernel functions can perform kernel trick such as linear (LF), radial basis (RBF), polynomial (PF), and sigmoid (SF) functions. The simplest kernel is LF, and only suitable when the observations are linearly related, RBF depends on the distance from the origin, PF is non-stationary and mostly suitable on normalized training data, and SF is based on neural networks (Pradhan et al., 2018).

After enlarging the feature space, the algorithm tries to determine an optimal hyperplane through maximization of safety margin from the nearest data points which are called “Support Vectors”. Such a hyperplane with maximized distance from support vectors is known as a “Maximum-Margin Hyperplane”. A traditional SVM model has been illustrated in Figure 3.6 This hyperplane is then utilized to classify features from an unknown dataset. The performance of SVM is highly dependent on its hyperparameters. In this study, the widely acclaimed RBF kernel was selected and three parameters were optimized to obtain the landslide inventory map: (1) cost penalty function (C) which contributes to margin maximization, (2) gamma (ϒ) indicates the curvature of the decision boundary, and tolerance (tol) is a stopping criterion. The general mathematical expression for RBF kernel function (Pradhan et al., 2018; Wang et al., 2020) is mentioned below:

$$K\ (x_i, x_{i'}) = \exp(-\Upsilon \sum_{j=1}^{m} \left(x_{ij}\ -\ x_{i'j}\right)^2) \quad \ldots\ldots (5)$$

$$\text{Where } \Upsilon = \frac{1}{2\sigma^2} \ldots\ldots (5a)$$

x_{ij} and $x_{i'j}$ indicates the ith observation pair for jth predictor, m is the number of predictors, ϒ is the decision boundary smoothness, σ is the variance, and K denotes the kernel function.

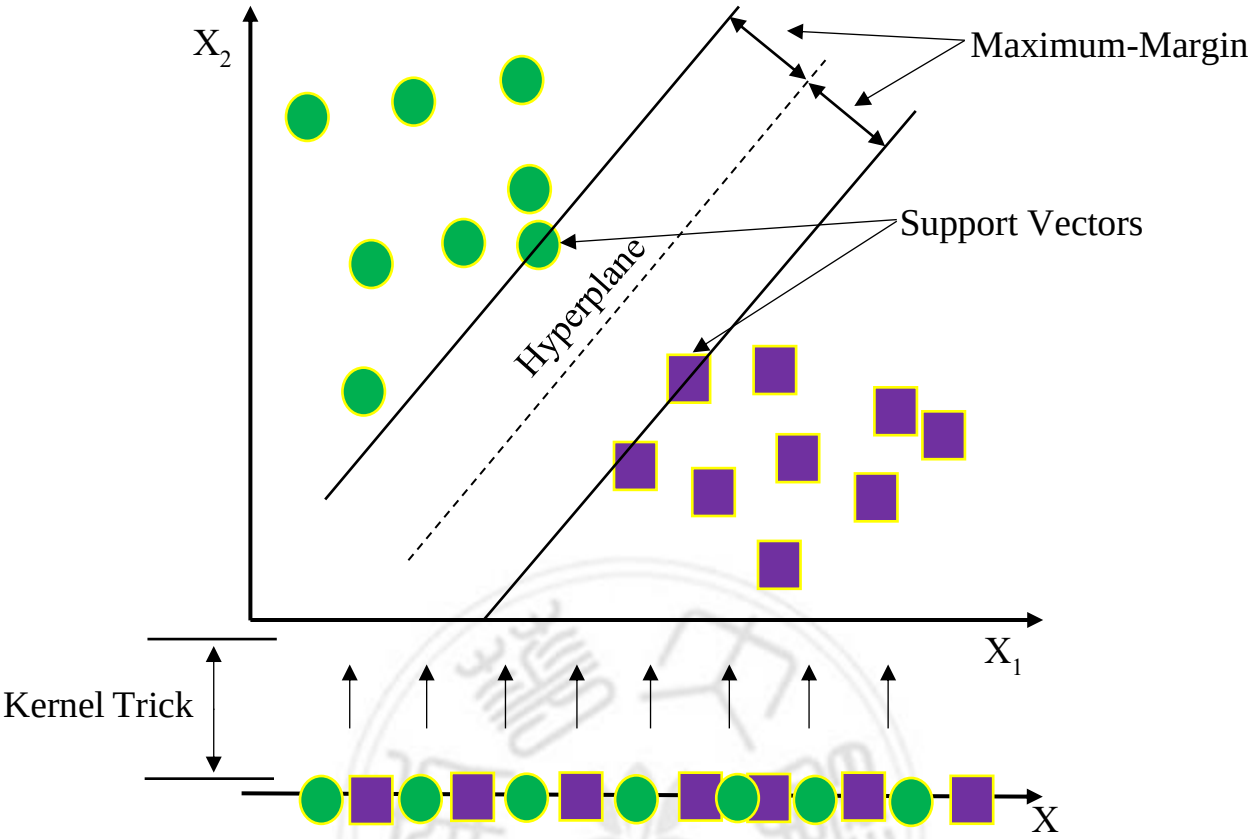


Figure 3.6: Representation of a Conventional SVM model

3.5.1.2 Decision Tree (DT)

A decision tree is an algorithm that is used to generate decision rules for classification and regression. The structural pattern of this algorithm is similar to tree structure which includes conditional criteria at each node for classification and regression problems. The decision tree is not sensitive to the relationship between all the input variables and the target variable (Alkhasawneh et al., 2014). It is a type of mapping algorithm that maps the rules and their conclusion to the target variable (Wan et al., 2010). The decision tree is not affected by input variables of different scales (Alkhasawneh et al., 2014), thus eliminating the step of data normalization from the data preprocessing. However, in this study, the decision tree was deployed on the normalized data for comparison. The primary objective of this algorithm is to

provide a concise and accurate representation of the relationship between the input variable and the target variable. Specifically, the decision tree can be easily visualized unlike “black box” algorithms such as neural networks.

The decision tree is a multilevel hierarchical decision structure and it mainly comprises of a root node, a set of internal nodes or child nodes, and at last end-nodes or terminal nodes (leaves), as shown in Figure 3.7. Each node is required to make a binary decision that either classifies one class or some other classes from the remaining ones. The computation is continued until a terminal node is reached. The length from the root node to a terminal node indicates the total depth of the tree. The decision tree selects input features with maximum information for classification, rejecting other remaining features to improve computational efficiency and time. There are four types of decision tree algorithms: “Classification and Regression Tree” (CART), “Chi-square Automatic Interaction Detection” (CHAID), “exhaustive CHAID”, and “Quick, Unbiased, and Efficient Statistical Tree” (QUEST). Studies from Park (2018) and Alkhasawneh (2014) established a detailed comparative study of these algorithms for landslide detection and susceptibility mapping, respectively.

For this study, the CART algorithm from the “scikit-learn” library was selected to generate a decision tree for landslide mapping. It is a repetitive segregation method and can be used for both classification and regression. The tree is generated by splitting subsamples to construct two child nodes from all predictors in a given dataset, starting with the entire dataset. The best predictor can be selected using various metrics such as Gini index, entropy, towing, ordered towing, least-square deviation. In this study, both Gini index and entropy were chosen for this purpose while the best criterion was automatically selected during the parameter tuning process.

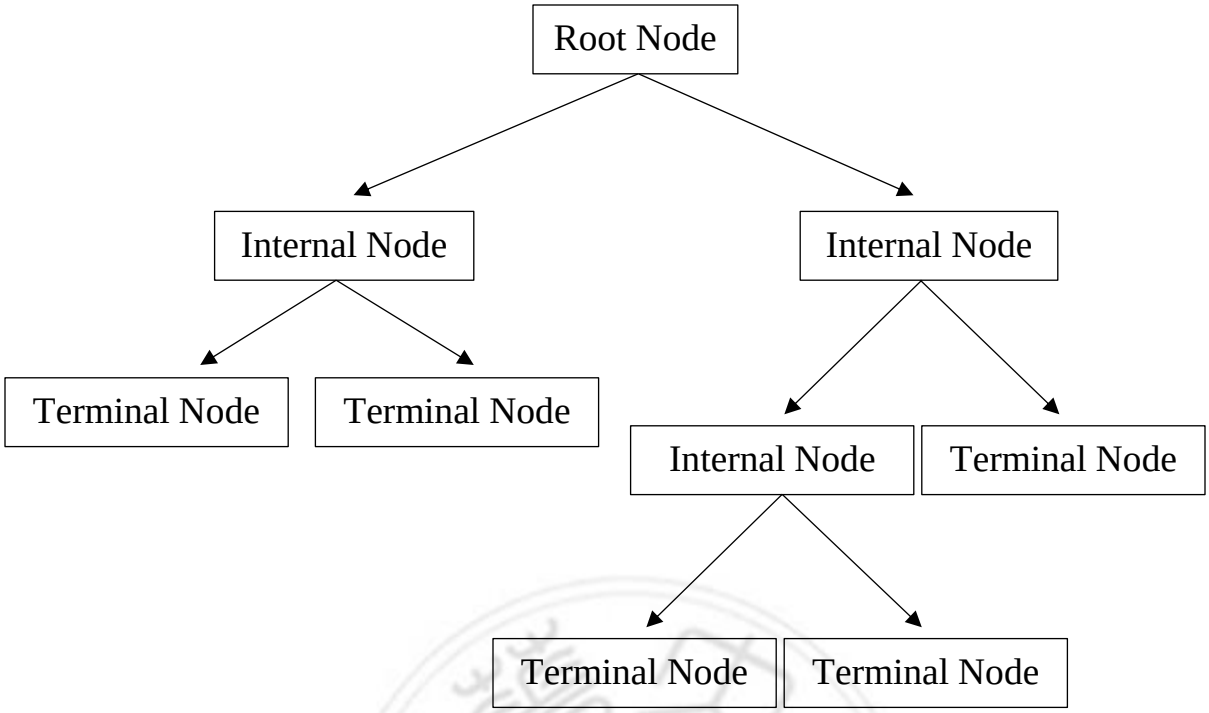


Figure 3.7: Typical Decision Tree Structure

Entropy can be defined as the impurity in input data [6]. While information gain (decrease in entropy) is the difference between entropy and the additional information required for an attribute (feature) which is the weighted average of the entropy in the dataset following the split. The mathematical expression for entropy (E(D)), new information (Info(D)), and information gain (Gain(D)) was given by Pradhan et. al. (2018). Consider a dataset D= {d_1, d_2, d_3, d_4, …, d_j}, the equation for entropy can be given by:

$$E(D) = -\sum_{i=1}^{n} p_i \log_2 p_i \quad \ldots\ldots (6)$$

Where, p_i is the probability for a portion in the dataset (D) belonging to class i. While the new information with weighted entropy is as follows:

$$Info(D) = \sum_{k=1}^{t} \frac{|d_k|}{|d|} \times E(D_k) \ldots\ldots (7)$$

In which, k represents the subset in dataset D= $\{d_1, d_2, d_3, \ldots, d_k\}$ while, the $|d_k|/|d|$ is the weightage of the k^{th} partition. $E(D_k)$ is the entropy for the k^{th} portion.

Finally, the information gain is calculated by Eq. (8)

$$Gain(D) = E(D) - Info(D) \ldots\ldots \quad (8)$$

Attribute having the highest information gain value will be selected for the split at the node (N).

Another, splitting criterion that is Gini index can also be implemented in the CART-based decision tree algorithm. The expression for Gini index (G(D)) for a dataset D is given by Alkhasawneh et. al. (2014) and specified in Eq. (9):

$$G(D) = 1 - \sum_{j=1}^{m} pi^2 \ldots\ldots \quad (9)$$

pi is the probability of subset in D belongs to class i. The attribute with the lowest Gini index value is used for splitting the node (N)

Five parameters were tuned for this algorithm: (1) "criterion" which measures the quality of split through a function, specified both Gini index and entropy, (2) "max_depth" where maximum depth of the tree is specified, (3) "max_leaf_nodes" which indicates the number of terminal nodes (leaves), (4) "min_samples_leaf" is the minimum data samples required at a leaf node, and (5) "min_samples_split" represents the number of samples required to split an internal node (6) "splitter" is the strategic choice for the split at each node (best or random), the optimal strategy for this research was "best" following the parameter tuning process.

3.5.1.3 Extra Trees Classifier (ET)

Extremely randomized trees classifier also known as Extra Trees Classifiers are a tree-based ensembled technique that utilizes results from de-correlated multiple decision trees to give classification results. It is quite similar to Random Forest classifiers except for its method of generating decision trees. Unlike the Random Forest (RF), the entire training dataset is used to generate each decision tree and the split is randomized, hence, less computational. The square root of the total features is utilized to generate each tree. The node-splitting for each decision tree is performed by determining the split through random selection of the features based on mathematical criteria (typically Gini index or entropy). The random sampling of features leads to the generation of multiple de-correlated decision trees. The final classification result is based on weighted majority voting by aggregating results from all the constructed decision trees. A simple representation of an Extra Trees classifier is illustrated in Figure 3.8. From the cited research, the Extra Trees Classifier had a very limited application on remote sensing data. However, it has been applied as one of the base classifiers for "Object-based Automatic Change Detection using Multiple Classifiers and Multi-scale Uncertainty Analysis" on high-resolution remote sensing images (Tan et al., 2019). Also, it has been deployed for detecting the open and vegetated water bodies from Sentinel-1 satellite images for mapping the African malaria vector mosquito breeding habitats (Hardy et al., 2019).

In this research, along with the five parameters mentioned earlier in section 3.5.1.2, one more parameter was optimized i.e., "n_estimators". This parameter represents the number of decision trees to be constructed for extra tree classification.

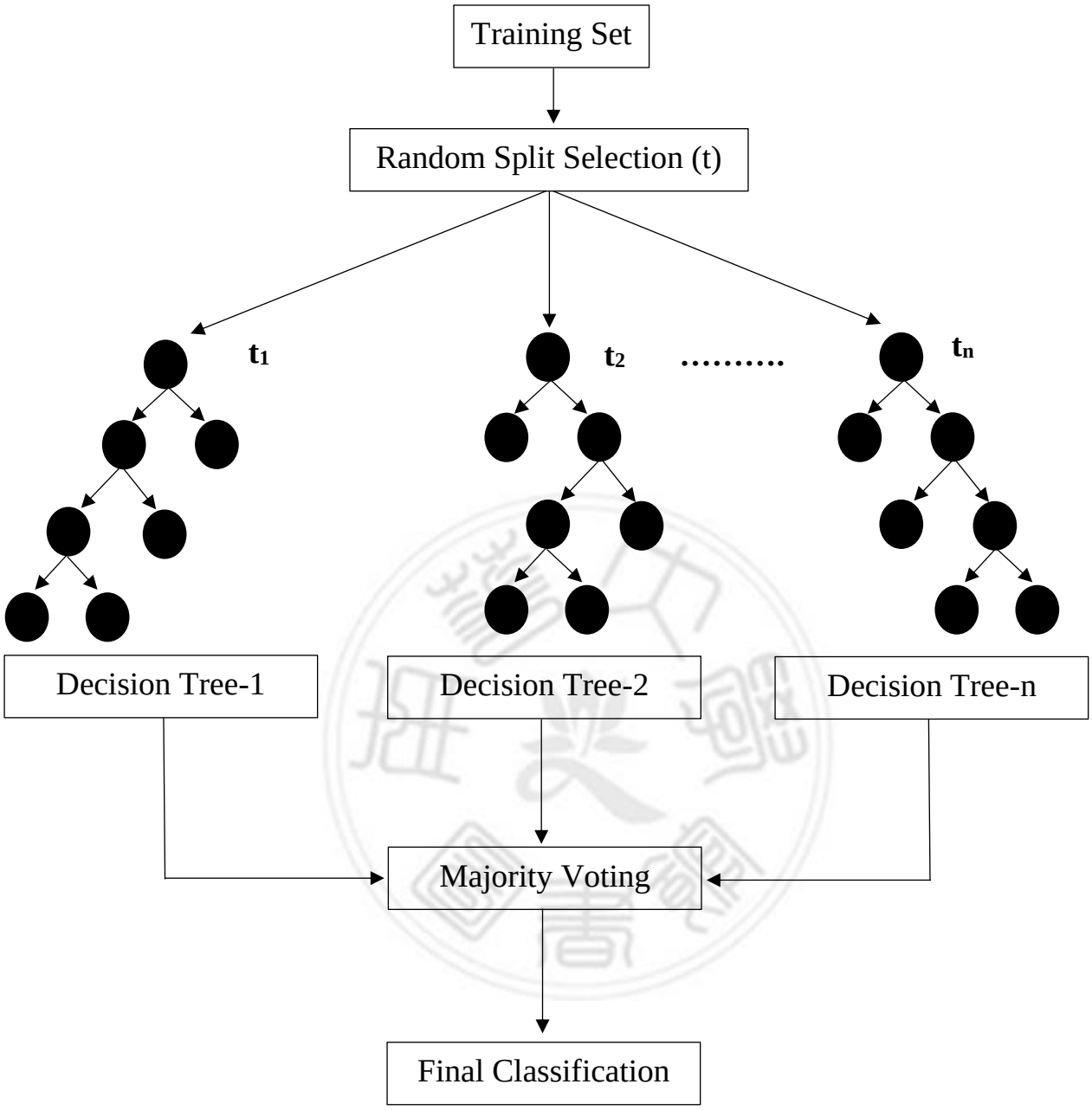


Figure 3.8: Representation of a Typical Extra Trees Classifier

3.5.1.4 Random Forest Classifier (RF)

Random Forest is a popular non-parametric, tree-based ensemble AI technique used for classification and regression problems. It has been widely deployed for the detection of landslides using remote sensing data (Chen et al., 2014;

Li, 2015; Pradhan, 2018; Ghorbanzadeh et al., 2019; Tavakkoli et al., 2019; Nhu, 2020; Wang et al., 2020). It provides reliable performance on a large and complex dataset with low computational cost. One of the major advantages includes no requirement for the assumption of the statistical distribution of the dataset (Pradhan et al., 2020). It is a widely popular tool to identify hidden patterns within a large volume of the dataset (Chen et al., 2014). Similar to Extra Trees, this technique also generates several de-correlated decision trees. However, this algorithm develops decision trees through random sampling of a given dataset instead of using the entire dataset by a technique known as bootstrapping, also it only chooses optimal split which is a little computationally costlier than Extra Trees. This algorithm is also not allowed to utilize all input features instead considers a random subset of input feature i.e., the square root of the total features. This algorithm is quite popular due to its less sensitivity to over-fitting as well as an acceptable performance from fewer efforts on parameter tuning. Similarly, the best split at each node is based on the mathematical criteria i.e., Gini index or entropy. The final classification result is acquired based on the majority votes across all the decision trees, as depicted in Figure 3.9. The cited literature indicated that the random forest classifier achieved highly accurate results on image classification. Therefore, the random forest has been selected for comparison purposes in this study.

Along with the aforementioned parameters in section 3.5.1.2 additional two parameters have been optimized for Random Forest i.e., the “n_estimators” that define the number of decision trees to be constructed, and “max_sample” which represents the fraction of samples to be utilized for constructing each decision tree.

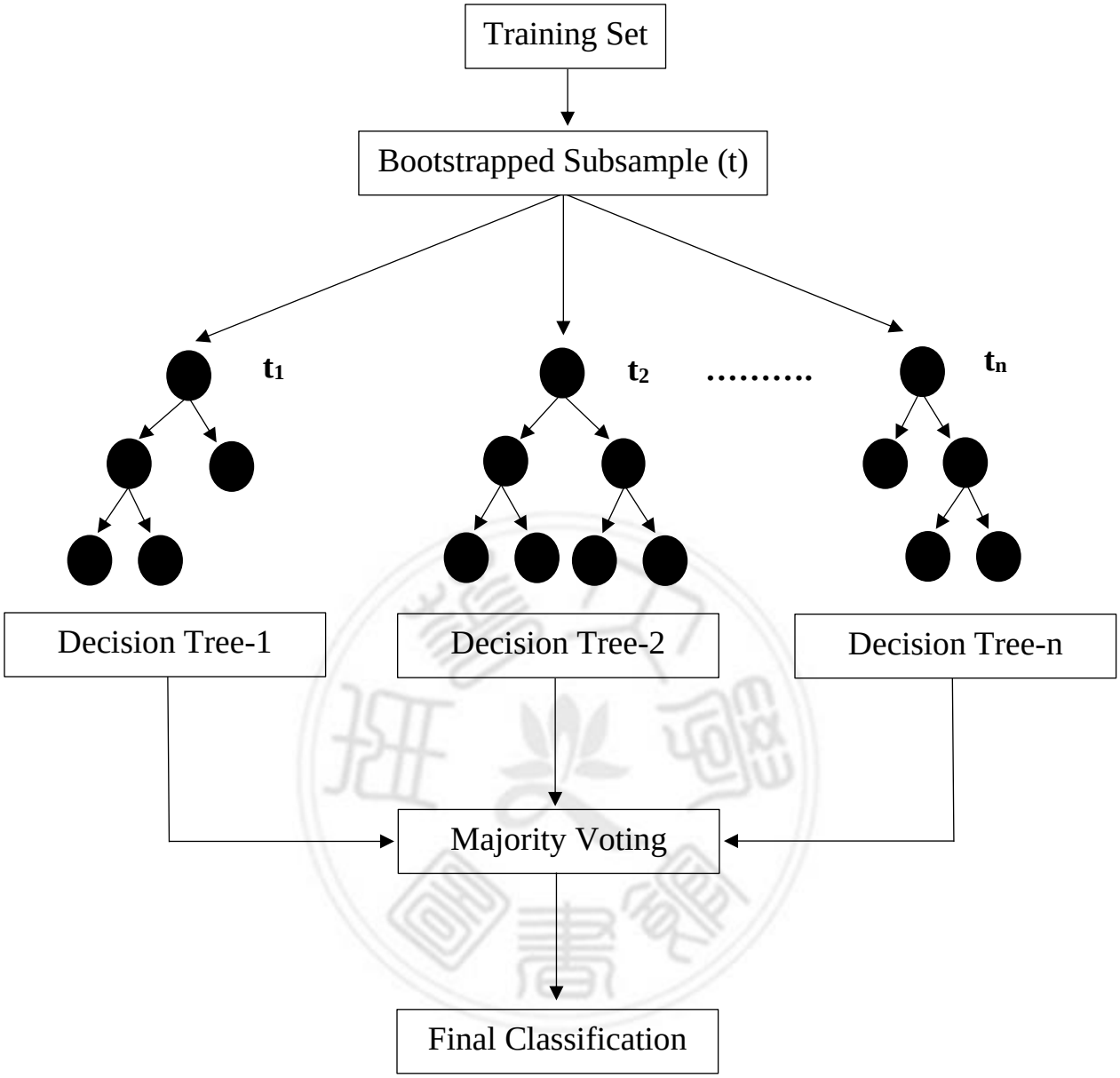


Figure 3.9: Typical Random Forest Classification

3.5.1.5 Extreme Gradient Boosting (XGBoost)

Another type of tree-based ensembled algorithm that constructs several weak learners, typically decision trees to assemble a powerful "committee", hence, regarded as a strong classifier (Wang et al., 2020). The algorithm generates weak learners by a process known as boosting, which is referred to as the training of weak

learners sequentially with each attempting to correct their predecessor (Ma et al., 2020) through weighted outcomes of the previous learner. This algorithm utilizes a regularized model formulation to avoid over-fitting, has a built-in feature to handle missing values, and compute cross-validation scores. It is also highly customizable and allows users to specify suitable evaluation criteria for their model (high flexibility). There are several other variants available based on gradient boosting strategy such as Adaboost, LightGBM, Catboost, etc. The Adaboost and especially XGBoost are some of the prominent boosting algorithms and have significantly produced reliable results for landslide detection and susceptibility mapping (Pradhan et al., 2020; Sahin et al., 2020; Wang et al., 2020). The XGBoost algorithm efficiently reduces the processing time by performing both classification and regression problems on multi-cores, hence, relatively faster than other boosting variants. The gradient boosting method consists of three main components: (1) a loss function that requires optimization, (2) a weak learner for prediction, and (3) an additive model that combines the weak learners to optimize the loss function (Sahin et al., 2020).

In this study, the XGBoost algorithm was chosen due to its wide adoption in past research works on landslides. The number of weak learners was mainly dependent on the Logarithmic Loss function (Log Loss), which penalizes inaccurate classification by considering its probability. The general expression for Log Loss function [32] is given in Eq (10):

$$Log\ Loss = \frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1 - y_i)\log(1 - p_i)] \quad \ldots\ldots\ldots (10)$$

N is the total number of samples, y_i is the outcome of ith instance, and p_i is the probability of the ith instance. Two evaluation metrics were monitored during the generation of weak learners: Classification Error, and Log Loss function. The

number of weak learners is generally referred to as epochs in this algorithm, which was given early stopping criteria if no reduction in Log Loss function was observed. A general description of the sequential tree construction based on boosting concept is given in Figure 3.10.

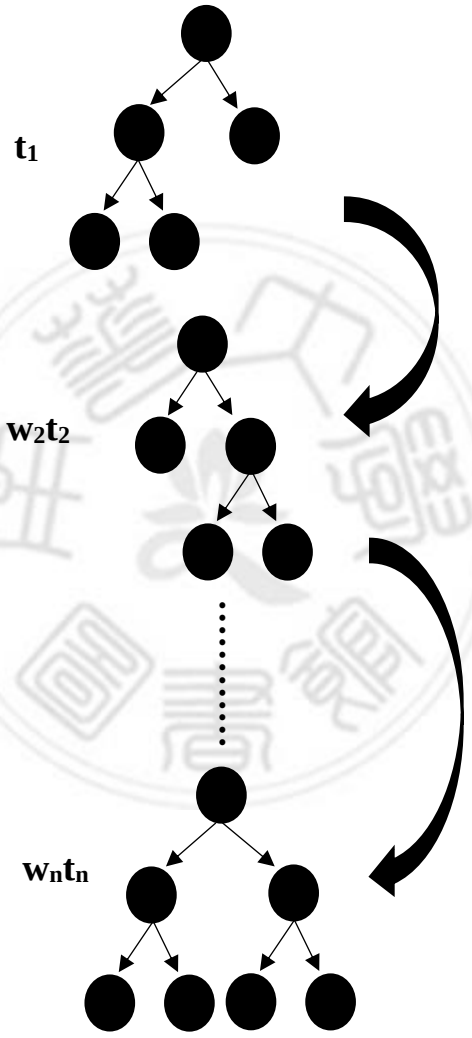


Figure 3.10: General Sequential Boosting Strategy for XGBoost

The XGBoost algorithm was implemented using an open-source code "xgboost" integrated with "scikit learn" API for python programming environment and was adjusted using nine parameters namely: (1) "booster" in which the method of boosting is specified, for this research "gbtree" was used for tree-based model, another option is "gblinear" for linear model, this parameter was not tuned, (2) "colsample_bytree" represents a portion of the columns to be randomly sampled for constructing one tree, (3) "gamma" is the minimum loss function for further partitioning of leaf node for the tree, (4) "eta" (learning rate) which is a step size shrinkage, after each step of boosting it shrinks the feature weights to make boosting more conservative, (5) "max_depth" which indicates the total depth of each tree, (6) "min_child_weight" defines the minimum sum of all weights required in a child node, (7) "reg_alpha" which is an L1 regularization, a coefficient of penalty term (absolute value) added in the loss function, (8) "reg_lambda" is the L2 regularization term, coefficient as a squared value is added to the loss function, (9) "subsample" indicates random sampling of training data before growing trees, and (10) "n_estimator" defines the number of maximum trees to be generated.

3.5.1.6 Adopted Training & Testing Methodology

The present study proposes to evaluate five supervised algorithms for landslide identification. Each algorithm was trained and tested on the pre-processed dataset as discussed in section 3.4. The performance or accuracy of the supervised learning algorithms significantly depends on the tuning of its parameters. This study adopts "GridsearchCV" from the "scikit-learn" library for python to determine the optimal values for each parameter. The predictive efficiency of these models was evaluated using various metrics such as "Test Accuracy" which gives an insight into the model's efficiency in classifying the testing set, "Cross-Validation" which is a resampling technique used to evaluate these models, and "Root Mean Square Error"

(RMSE) is a measure of the difference between the prediction of the model and actual observation. If the difference between RMSE for the training set and testing set is too large then the model is badly fit on a given dataset. Similarly, the difference between Test Accuracy and the Cross-Validation also indicates the fitting condition i.e., overfit or underfit of the trained model. Each of these metrics is discussed in detail subsequently. For Extreme Gradient Boosting (XGBoost), two additional metrics were visualized to check the quality of fit by weak learners on the given training and testing sets i.e., Classification Error and Logarithmic Loss function (Log Loss) for each epoch.

(1) Cross-Validation

It is a statistical method for partitioning samples from a given data into several subsets that performs analysis on a single subset, while the remaining subsets are consequently reserved for the validation and confirmation of the initial analysis (Wan et al., 2010). This study adopts K-fold Cross-Validation for evaluating the actual dataset. Initially, the dataset was subsampled into K partitions, from which a single subsample is used as training data for training the model, while the remaining K - 1 subsamples are used for the testing. The cross-validation is repeated for K times (folds), where each of the K subsamples was used exactly once as training data. The mean of K-iteration scores can be used to determine the overall generalization of a model on the given dataset. For this study, the mean score from 20 iterations (folds) was utilized to acquire the overall cross-validation score for each model.

(2) Root Mean Square Error (RMSE)

It is the standard measure of predictive accuracy of a model in quantitative data. It is a measure of error between predicted data and observed data (Nhu et al., 2020). It can be computed by Eq (11):

$$RMSE = \sqrt{\frac{1}{i}} \sum_{i-1}^{i} \left(Y_{predicted} - Y_{actual}\right)^2 \quad \ldots\ldots (11)$$

Where i defines the total training & testing dataset, $Y_{predicted}$ and Y_{actual} represent the predicted and actual label of the training & testing set, respectively.

In this study, the root mean square error was calculated for both training and testing set to evaluate the fitting condition of the trained model.

(3) Hyperparameter Tuning

As mentioned earlier, the classification performance of the supervised learning algorithm is significantly dependent on the tuning of its parameter. GridsearchCV, which is a popular parameter tuning technique was utilized to tune each parameter of these models. It is a method of determining the optimal values for each parameter by looping over a grid of values specified by the user. This method investigates all the probable combinations from each parameter to obtain the best combination of optimal values. The parameters were tuned until the model satisfies the following criteria of each evaluation metrics specified in the previous sections:

(a) The difference between Test Accuracy and Cross-Validation score can indicate whether the trained model is overfitting or underfitting. If the variation is too large i.e., Cross-Validation higher than Test accuracy then the model is said to be overfitting, and vice-versa for the underfitting. The parameters were tuned until the variance was minimal.

(b) The Root Mean Square Error (RMSE) also indicates the fitting condition of the proposed model. If the difference between the RMSE of the training set and the testing set is too large, it indicates the model cannot fit properly on the given dataset. Parameter tuning continued until a minimal difference was obtained.

(c) For Extreme Gradient Boosting (XGBoost), the model was initially set to generate 1000 learners (epochs) with an early termination criterion, which was kept at 10 rounds indicating that the training process automatically terminates if no reduction was observed in the Log Loss function for the next 10 epochs.

3.5.2 Unsupervised Approach

In this modern era, the requirement of a rapid, robust, and efficient hazard assessment system is critical for emergency responses. As the name suggests, unsupervised algorithms do not require prior training or testing on a labeled dataset instead, quantifies the whole data into pre-defined clusters or classes with no prior human supervision. This fulfills the requirements of a rapid monitoring system post-natural hazard. As discussion on the advantages of remote sensing data over traditional methods for hazard monitoring is rapidly increasing, some research works have successfully implemented unsupervised techniques with satellite images for landslide detection (Keyport et al., 2018; Lei, 2018; Ramos-Bernal, 2018; Tran et al., 2019; Zhai et al., 2020). This study proposed to evaluate four unsupervised algorithms, in which K-means, Minibatch K-means, BIRCH, and Gaussian Mixture Models (GMM) were used to generate topographic signatures of the landslides. Each of these algorithms is implemented through the open-source library “scikit-learn” for the python environment. Subsequently, each algorithm is discussed in detail.

3.5.2.1 K-means

The K-means clustering algorithm is a partitioning-based clustering technique that is used for a variety of applications. There are successful implementations of K-means with remote sensing images for landslide detection. Tran et. al. (2019) successfully derived landslide maps through K-means using airborne laser scanning data (LiDAR). Sathiaraj (2019) and Cintia (2020) indicated that it's crucial to select initial cluster centers to acquire high accuracy. K-means is employed on a wide variety of datasets for its simplicity, flexibility, and efficiency. A simple depiction of the clusters computed through K-means is illustrated in Figure 3.11. The K-means algorithm calculates clusters over several iterations. The iterations are continued until a clustering criterion is optimized. The sum of squared Euclidean distance is a commonly used criterion for this purpose. Given a dataset D = {d_1, d_2, ……, d_i}, where i is the number of points to be clustered into k clusters. Assuming cluster as C = {c_1, c_2, ….., c_k}, the summation of squared Euclidean distance between points d_i is computed for the cluster centroid O_k of the subset c_k, which contains d_i. This criterion is known as clustering error (E) and corresponds to cluster centroids O_1, O_2, ….., O_k (corresponds to pre-defined number of clusters). The mathematical expression for E (Sathiaraj et al., 2019) corresponding to each cluster c_k is given by Eq (12):

$$E(c_k) = \sum_{d_i \in c_k} ||d_i - O_k||^2 \quad \text{…….. (12)}$$

The algorithm focuses on minimizing the squared errors across all clusters. It is a type of iteration-based algorithm, that randomly selects initial clusters and iteratively reassigns cluster centers until the squared error between data points inside each cluster is minimal. As mentioned earlier, the main disadvantage of this algorithm is its dependency on the selection of the initial position of the clusters to acquire optimal results. For this research, the initial random positioning of clusters

was implemented through parameter “init” in which the method of initiation is specified. The initiation method specified was “k-means++” which speeds up the selection process of initial cluster centers in a smart way to speed up the convergence. In this study, 50 clusters were specified with max iterations of 100.

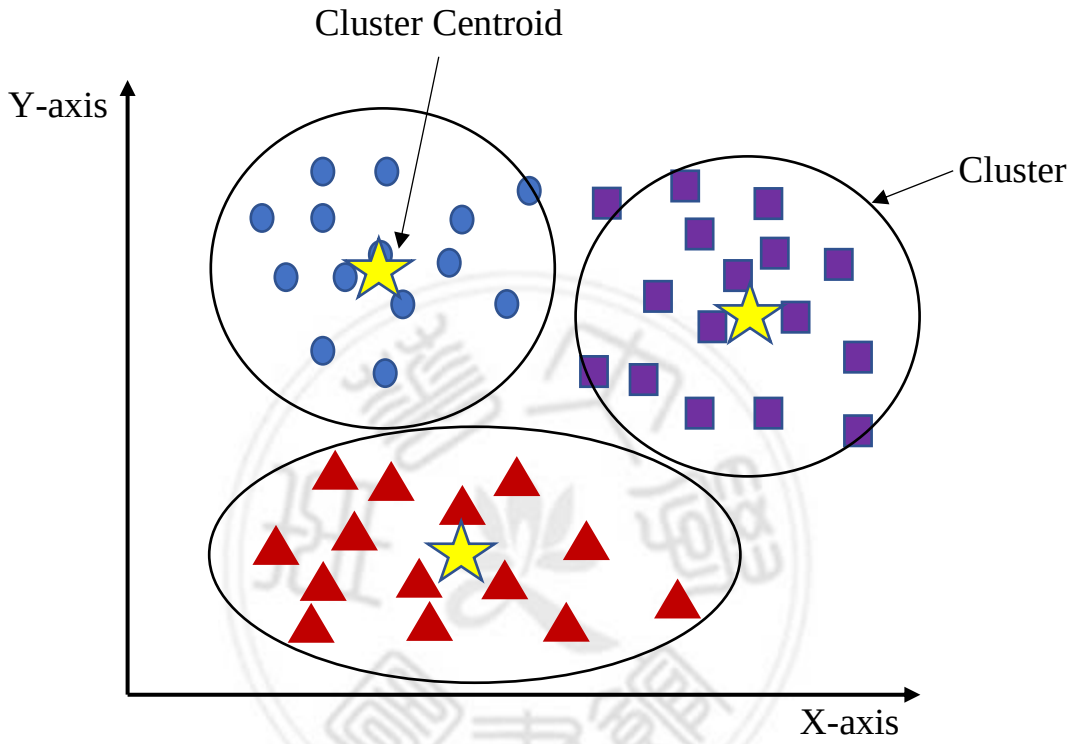


Figure 3.11: K-mean Clusters

3.5.2.2 Minibatch K-means

A modified version of the K-means algorithm was developed i.e., Minibatch K-means, which reduces the time and complexity of large-scale calculations in huge datasets (Cintia et al., 2020). The reduction of inertia and the sum of squares within clusters are the primary goal of this algorithm (Vergunst, 2017). As per the cited literature, the application of Minibatch K-means is fairly limited with remote sensing data. Although, the algorithm was employed for clustering hyperspectral remote sensing images (Reddy et al., 2020), and also a CNN-based Minibatch K-means for

joint clustering was proposed to represent large-scale image data (Hsu et al., 2017). The algorithm first divides the input data into various subsamples or mini-batches, that are individually used and significantly shortens the computation time (Vergunst, 2017). In the first step, the subsamples are randomly chosen to generate mini-batches followed by allocation to the nearest centroid. The second step involves updating the centroids for all generated mini-batches based on clustering criterion which is squared Euclidean distance as discussed in section 3.5.2.1. Each centroid gets updated based on the streaming average of a specific batch used in that iteration (Vergunst, 2017). The algorithm iterates until convergence is achieved which can be the optimal clusters or the number of pre-specified iterations is reached. For a given dataset, D = $\{d_1, d_2, \ldots\ldots, d_i\}$, $d_N \in R^{p,q}$, where d_N indicates the network record of q-dimensional real vectors, whereas p represents the number of data points inside the dataset D. The goal is to compute the set of cluster centers $c \in R^{p,q}$ to minimize the given dataset D of clusters $c \in R^{p,q}$ in function. A general mathematical expression (Cintia et al., 2020) to compute the squared distance of one cluster is given by Eq (13):

$$\min \sum_{d \in D} \left|\left| f(C, d) - d \right|\right|^2 \quad \ldots\ldots. (13)$$

Where $f(C, d)$ is the nearest cluster center $c \in C$ for a datapoint d, and $|C| = K$, in which K is the required number of clusters.

In this study, seven parameters were specified for Minibatch K-means i.e., the number of clusters is specified to be 50 (same as K-means), “init” was k-means++ which is a smart selection of initial cluster for fast convergence, maximum iteration was specified as 150, “max_no_improvement” is an early stopping control for the subsequent portion of mini-batches that do not yield any improvement, the specified value was 10, “tol” is another early stopping criteria based on changes to cluster

centers after each iteration, given value was 0.001, and “reassignment_ratio” which controls the fraction of the maximum number of counts for a center to be reassigned, given value was 0.1. These parameters were not optimized and were only assigned based on the user’s experience and knowledge.

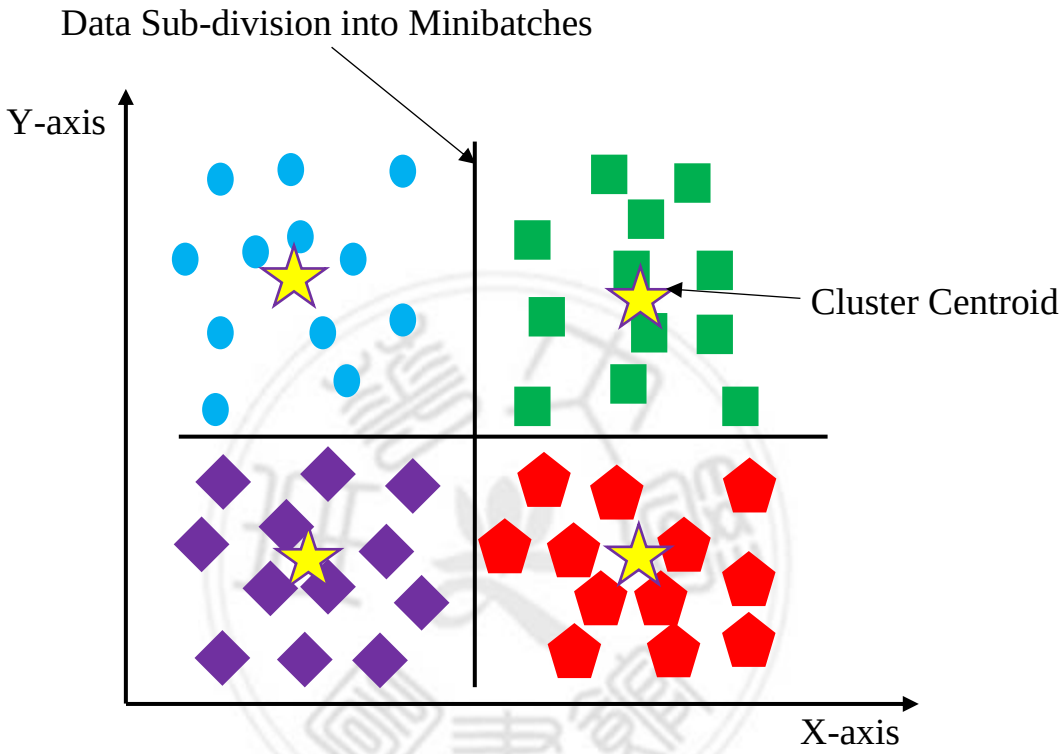


Figure 3.12: Clusters from Minibatch K-means

3.5.2.3 Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)

BIRCH is a type of hierarchical clustering algorithm that skims the whole dataset and compacts or summarizes it assuming that not all the data points are important. The summarized dataset retains the original information as much as possible and generates an in-memory tree representation of the clusters (Sathiaraj et al., 2019). This algorithm is highly suitable for clustering large datasets similar to Minibatch K-means. The basic principle of this algorithm is to develop clustering features (CF) for each cluster and builds a tree representation known as CF-tree. For

clustering, this algorithm includes four steps to achieve its result. The first step involves the creation of a CF-tree from the available dataset (Vergunst, 2017). Clustering features are triplets consisting of data points (D), linear sum (LS), and square sum (SS) of all data points in the dataset (Vergunst, 2017; Sathiaraj et al., 2019; Ren, 2020; Cintia et al., 2020), it can be represented by the set as CF = (D, LS, SS), where

$$\overrightarrow{LS} = \sum_{i=1}^{D} \overrightarrow{d_i} \quad \ldots\ldots (14)$$

$$\overrightarrow{SS} = \sum_{i=1}^{D} \overrightarrow{(d_i)^2} \quad \ldots\ldots (15)$$

The assembly of cluster features into CF-tree is based on two factors: branching factor (B) which is the maximum number of subclusters in each internal node and threshold (T) is the specified number of data points inside each sub-cluster in a leaf node, as shown in Figure 3.13. All the nodes contain mostly B subclusters, and these entries are formed by CF and child nodes. These child nodes contain CF itself restricted by a certain amount (B), resulting in a tree consisting of CF's. In the second step, the algorithm analyses the developed tree and tries to create a smaller tree by removing outliers and combining similar CF's. The third step involves clustering of leaves or child nodes of the CF. By compression, the algorithm achieves the pre-specified number of clusters by the user. The fourth or final step involves error identification and redistribution of the data (Vergunst, 2017).

As indicated by the cited literature & survey, there is a little-to-no application of the BIRCH algorithm on remote sensing data let alone for landslide identification. Although, it was applied to predict climate types using temperature and precipitation data for the continental United States (Sathiaraj et al., 2019). It has also been applied for the partitioning of dimensionally reduced hyperspectral images (Ren et al.,

2020). For this study, the parameters branching factor (B) and threshold (T) were kept at 50 and 0.01, respectively.

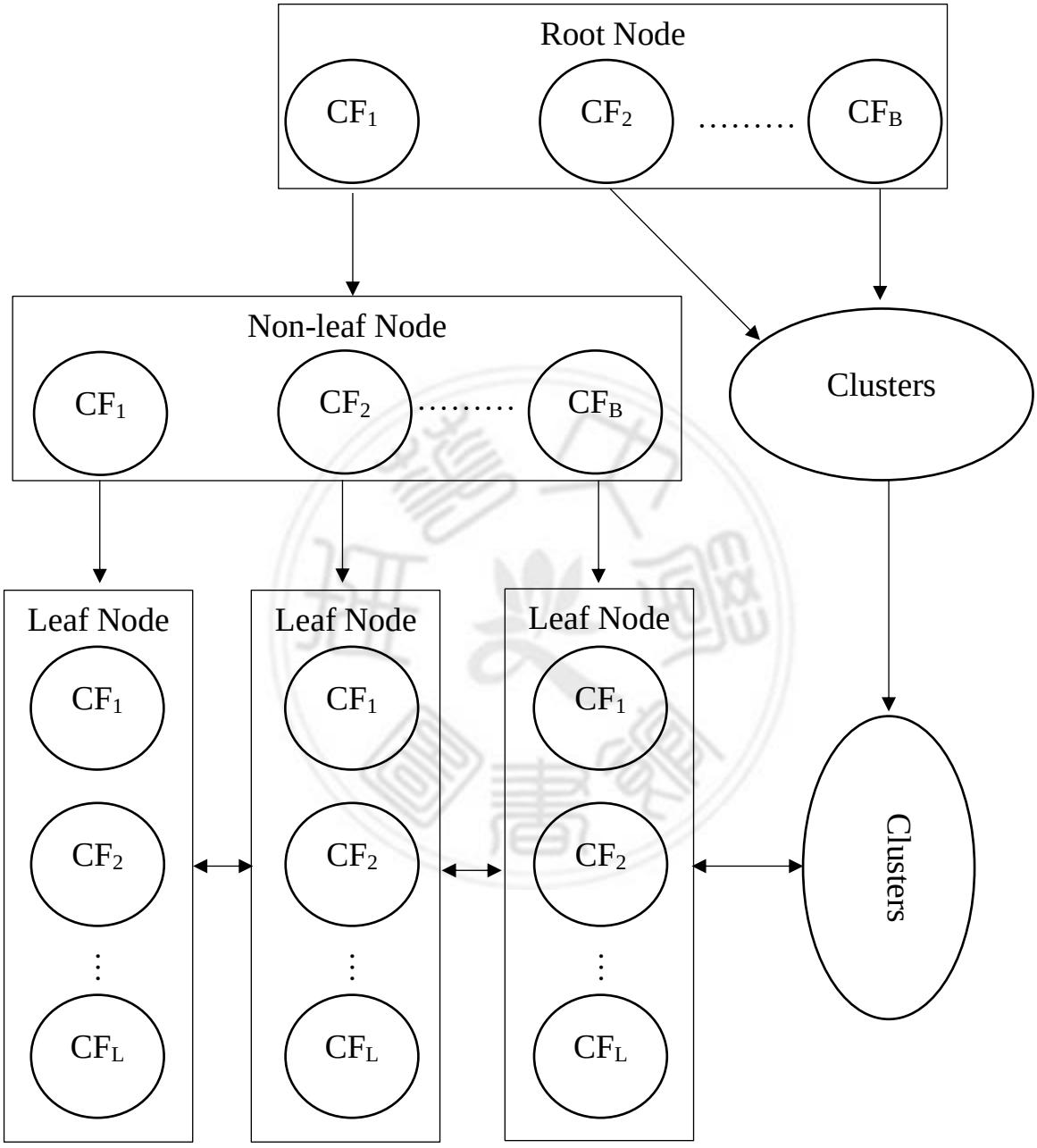


Figure 3.13: Clustering using Hierarchies (BIRCH)

3.5.2.4 Gaussian Mixture Models (GMM)

It is a probability model type of algorithm that assumes Gaussian distribution for all data points (Vergunst, 2017). GMM clusters maximize the hypothesis that a statistical probability is true, and relative to the dataset (Tran et al., 2019). Recently, the GMM clustering algorithm was employed on LiDAR-derived DEM for landslide detection and was compared with K-means (section 3.5.2.1) (Tran et al., 2019). Ari & Aksoy (2010) suggested "Particle Swarm Optimization" for GMM to classify satellite images, whereas Li (2013) employed GMM in conjunction with "Markov Random Fields" for hyperspectral image classification. GMM is advantageous for being efficient and flexible. The availability of hard and soft/fuzzy clustering makes it highly flexible (Vergunst, 2017; Tran et al., 2019). Hard clustering is assigning a data point strictly to one cluster, whereas soft/fuzzy clustering assigns probability scores to each data point for its belongingness to a particular cluster (Tran et al., 2019). Some drawbacks include the requirement of a user-defined number of clusters for fitting and the assumption of the dataset being normally distributed (Gaussian distribution).

The clustering process for this algorithm involves the assumption that there is a certain number of Gaussian distributions in the dataset and each of these distributions is described as a cluster. This algorithm tends to cluster data points based on the probability of their belongingness to a particular distribution while parameters are unknown as depicted in Figure 3.14. The parameters of the model are estimated by Expectation-Maximization (EM) (Li et al., 2013), which counters the ill-effects caused by the missing values or incomplete dataset. The model first generates training data by using the EM algorithm. It employs two steps where the first step is to generate expected values which are then maximized to be used in the next step. The primary focus here is to maximize the likelihood of the parameters,

which can be iterated over many times. The algorithm can converge to a local optimum while maintaining a reasonably fast algorithm (Vergunst, 2017). After the model is generated based on the given parameters, which can also be modified later, the model is then deployed on the rest of the data for clustering.

For GMM, a probability density function for likelihood was given as the sum of K Gaussian components (Li et al., 2013), which is given by:

$$P(X|\omega) = \sum_{k=1}^{K} \alpha_k \eta(X, \mu_k, \xi_k) \ldots\ldots\ldots (16)$$

Where, $\eta(X, \mu_k, \xi_k)$ indicates k^{th} Gaussian component, K represents the total number of mixture components (clusters), α_k, μ_k, ξ_k define the mixing weight, mean, and covariance matrix, respectively.

In this study, the number of components specified was 50 and all other available parameters in the "scikit-learn" library for GMM were kept at default.

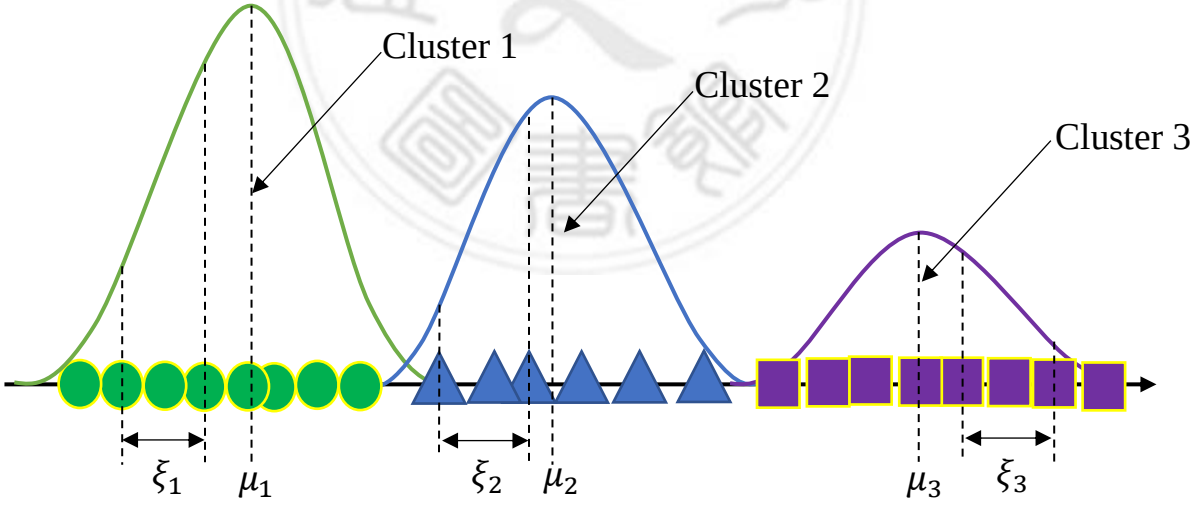


Figure 3.14: Clusters based on Gaussian Distribution

3.5.2.5 Adopted Clustering Methodology

As discussed earlier, landslide inventory maps were generated using four unsupervised clustering algorithms. The parameter selection is quite limited in unsupervised algorithms, which could assist in fast emergency response for inventory mapping (Zhai et al., 2020). The number of clusters can be manually selected by the user. The obtained differenced images with two input features (NDVI and TBI) were first normalized and then clustered into 50 distinct centroids, followed by subsequent manual recoding of the obtained image into landslide and non-landslide regions.

3.6 Quantitative Analysis of the Landslide Inventory Maps (LIM)

Assessment of landslide inventory maps acquired from unsupervised and supervised pixel-based classification requires a validation set with precise ground truth information (Keyport et al., 2018). Assessment of landslide maps from the validation set contains a total of 52 polygons, in which 40 polygons (1247 pixels) corresponds to verified landslide regions and 12 large polygons (1436 pixels) represents regions identified as non-landslide. These reference pixels were verified from the past research works conducted by Lin et. al. (2004) & Lin et. al. (2006). These points were compared with the landslide inventory maps produced by supervised and unsupervised algorithms to obtain omission and commission errors for each category.

The inventory map assessment was conducted by the error matrix, which was used for comparison. Metrics such as “Producer’s Accuracy” (PA), “User’s Accuracy” (UA), “Kappa Statistics” (K), and “Overall Accuracy” (OA) were derived from the error matrix.

3.6.1 User's Accuracy (UA)

It is an accuracy of a map based on the perspective of a map user, not the map maker. It is the ratio of the total number of correctly classified pixels in a category and total classified pixels in that category (Row Total).

It represents the possibility of a pixel that was classified by the model into a given group actually defines that land cover group on the field.

$$UA = \frac{Number\ of\ Correctly\ Classified\ Pixels}{Row\ Total} \times 100 \quad \ldots\ldots (17)$$

3.6.2 Producer's Accuracy (PA)

The Producer's Accuracy is based on the map maker instead of the map user. It can be calculated by dividing the total number of correctly classified pixels in a given category by the total number of reference pixels in that category (Column Total).

It defines how well the pixels in the validation set are correctly classified in the given land cover category.

$$PA = \frac{Number\ of\ Correctly\ Classified\ Pixels}{Column\ Total} \times 100 \quad \ldots\ldots (18)$$

3.6.3 Overall Accuracy (OA)

Overall accuracy is the total percentage of correctly classified pixels. It can be computed using the total number of correctly classified pixels divided by the total number of reference points.

$$OA = \frac{Total\ Number\ of\ Correctly\ Classified\ Reference\ Pixels}{Total\ Number\ of\ Reference\ Pixels} \times 100 \quad \ldots\ldots (19)$$

3.6.4 Cohen's Kappa (K)

Also known as Kappa Statistics or Kappa Index or just Kappa is much more robust than the conventional percentile-based accuracy metrics. The Kappa coefficient indicates whether the probability of an agreement is by chance or is it random.

The Kappa Statistics is denoted by K and can be calculated as:

$$K = \frac{(N \times d) - q}{N^2 - q} \quad \ldots\ldots (20)$$

Where N is the total reference points, d is the total sum of correctly classified pixels, while, q is given by Eq. (20a)

$$q = (Row\ Total\ \times Column\ Total) + (Row\ Total\ \times Column\ Total) \ldots\ldots (20a)$$

Non-Landslides	Landslides

The aforementioned metrics were computed from the error matrix of all nine landslide inventory maps as per the mathematical formulations specified above and compared. The model producing a landslide map with the best overall assessment was selected as the finest model. The next chapter reports the results obtained from all the proposed algorithms.

Chapter 4 Results

In this chapter, the results are presented that were obtained from the benchmarking tools to evaluate the training of supervised algorithms. Also, a detailed comparison is illustrated for the identified non-landslide and landslide regions to determine the best algorithm for generating a landslide map with the highest classification accuracy.

4.1 Landslide Inventory Maps (LIM)

4.1.1 Supervised Approach

The dataset as described in the previous chapter was used for the training and testing phase for supervised machine learning. Various benchmarks such as Test Accuracy, 20-fold Cross-Validation, and Root Mean Square Error were computed to examine the fit of the trained models on the established dataset. Subsequently, the trained models were used to classify and generate the Landslide Inventory Maps.

The acquired dataset outlined in the previous chapter was subdivided into a training (25%) and testing set (75%). The algorithms were trained and then the fit condition on the given dataset was evaluated using three metrics: (a) Test accuracy, the classification score on the testing set, (b) K-fold Cross-Validation score, the mean accuracy for 20 folds, (c) Root Mean Square Error, computed for both training set and testing set. For the parameter optimization, “GridsearchCV” was employed to obtain optimum values for each parameter. This technique requires a set of probable values for each parameter specified by the user’s knowledge and experience in the field of machine learning. The technique looped over all the possible combinations of pre-specified values and returned the most optimal combination of values that provided the highest classification score. The benchmark set for training the model and optimizing its parameters are as follows:

(a) The variation between Test Accuracy and 20-fold Cross-Validation accuracy was targeted to be kept below 5% (0.05), as a large variation may result in overfitting or underfitting of the model.

(b) The difference between RMSE values for the training and testing set was kept below 0.05.

Table 1: Values for the Evaluation Metrics

Algorithms	Test Accuracy	Cross-Validation	RMSE for the Training Set	RMSE for the Testing Set
SVM	0.9639	0.9686	0.1763	0.1899
DT	0.9659	0.9717	0.1532	0.1844
ET	0.9679	0.9719	0.1655	0.1791
RF	0.9676	0.9730	0.1574	0.1799
XGBoost	0.9685	0.9721	0.1595	0.1772

The best aforementioned metric scores obtained following the training and parameter tuning for each algorithm are specified in Table 1. All the algorithms showed better variation than the target benchmark. SVM and ET showed the same and lowest variance in RMSE of 0.0136 for both sets followed by XGBoost with 0.0177. The RF model showed an average RMSE variance of 0.0225, the highest variance was observed for the DT model of 0.0312. The lowest difference between Test Accuracy and Cross-Validation was observed in the XGBoost model at 0.0036. The ET model nearly follows XGBoost with a variation of 0.0040 followed by SVM with 0.0047. The highest variance was observed in DT and RF of 0.0058 and 0.0054, respectively. The fine-tuning of each algorithm was conducted until the best minimal variation was achieved. However, the borderline variance was kept at 5% between Test Accuracy and Cross-Validation, whereas 0.05 for RMSE. If any of the proposed models showed variance larger than this then that respective model would be

subsequently rejected from this study. All models showed good results in terms of benchmark metrics proposed in this study for training and parameter tuning purposes. Although, low variation doesn't automatically guarantee a superior classification accuracy for developing predictive systems, but only indicates how well the model has fitted on the given dataset.

Additionally, two more evaluation metrics were monitored during the training of the XGBoost algorithm. For this algorithm number of weak predictors are constructed sequentially in which each new predictor is an improvement over its predecessor. Classification Error and Log Loss function as described in section 3.5.1.5 was monitored for each constructed predictor on training and testing set simultaneously. Too many weak learners may tend to make the XGBoost algorithm overfit. So initially, the number of predictors was defined at 1000 with an early stopping criterion. The stopping criteria was kept for 10 rounds, indicating that the sequential generation of weak predictors will be terminated if no improvement (reduction) in the Log Loss function in only the testing set was observed for the next 10 learners (epochs). During the training process, classification error decreased with minor fluctuations until it reached 203 epochs after which classification error for the testing set started to gradually increase, whereas for the training set it was nearly constant. The Log Loss function sharply declined till round 124 after which negligible deviations were observed resulting in a near about constant straight line as shown in Figure 4.2. The training process terminated at round 213 with an output statement indicating 203 as the best early stopping round. At the first epoch, the classification error was 0.08159 and 0.09229 whereas the Log Loss function value was 0.65895 and 0.65984 for the training and testing set, respectively. Both simultaneously reduced at 203 epochs with an error value of 0.02546 and 0.03141, and the loss value was at 0.08969 and 0.10785 for the training and testing set,

respectively. As per the customized criterion, the Log Loss function was monitored for only the testing set. From epoch 204 the loss function marginally rose and the training was terminated at round 213 where the loss function markedly increased to 0.10806 from 0.10785 at 203. Hence, these 203 weak learners were chosen as the best committee of weak learners for detecting landslides. Graphs were plotted to visualize the change in Classification Error and Log Loss function for each epoch till early termination are presented in Figures 4.1 and 4.2, respectively. The optimal parameter values for each algorithm are given in Tables 2, 3, and 4.

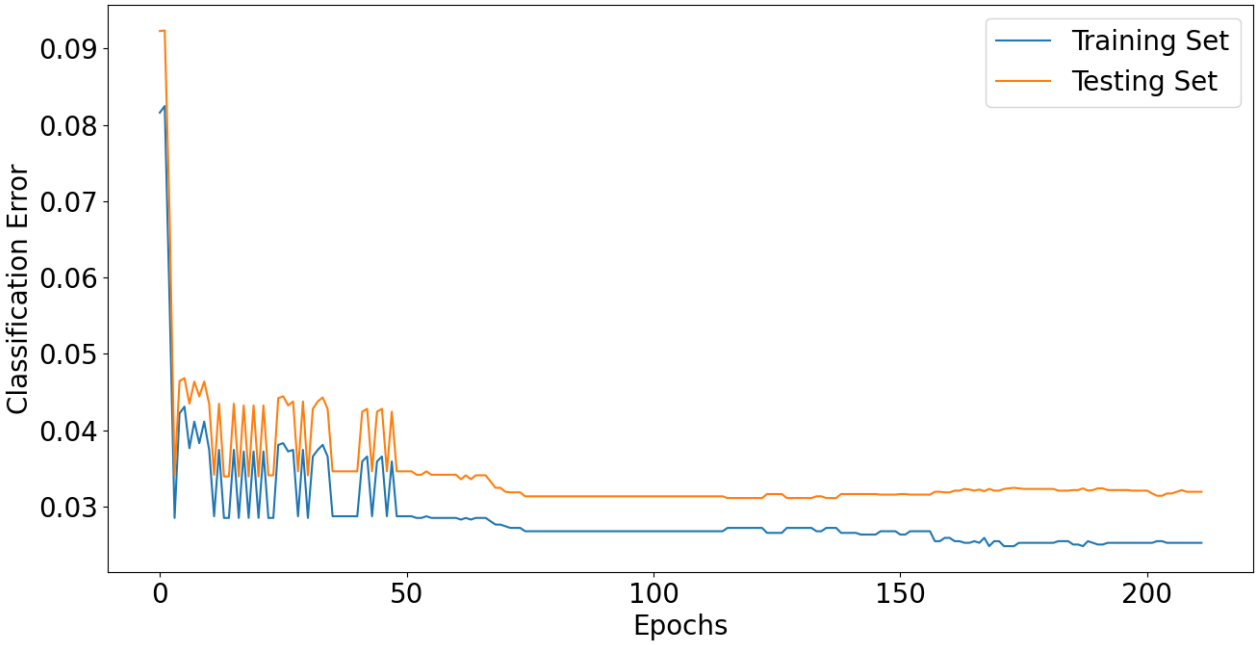


Figure 4.1: Plot for XGBoost Classification Error

Table 2: Optimal Parameter values for Ensembled Tree Algorithms

Algorithm	Criterion	Max Depth	Max Leaf Nodes	Max Samples	Min Samples Leaf	Min Samples Split	No. Of Estimators
DT	Entropy	19	39	-	3	2	1
ET	Gini	17	45	-	2	2	84
RF	Gini	7	36	0.9644	3	7	105

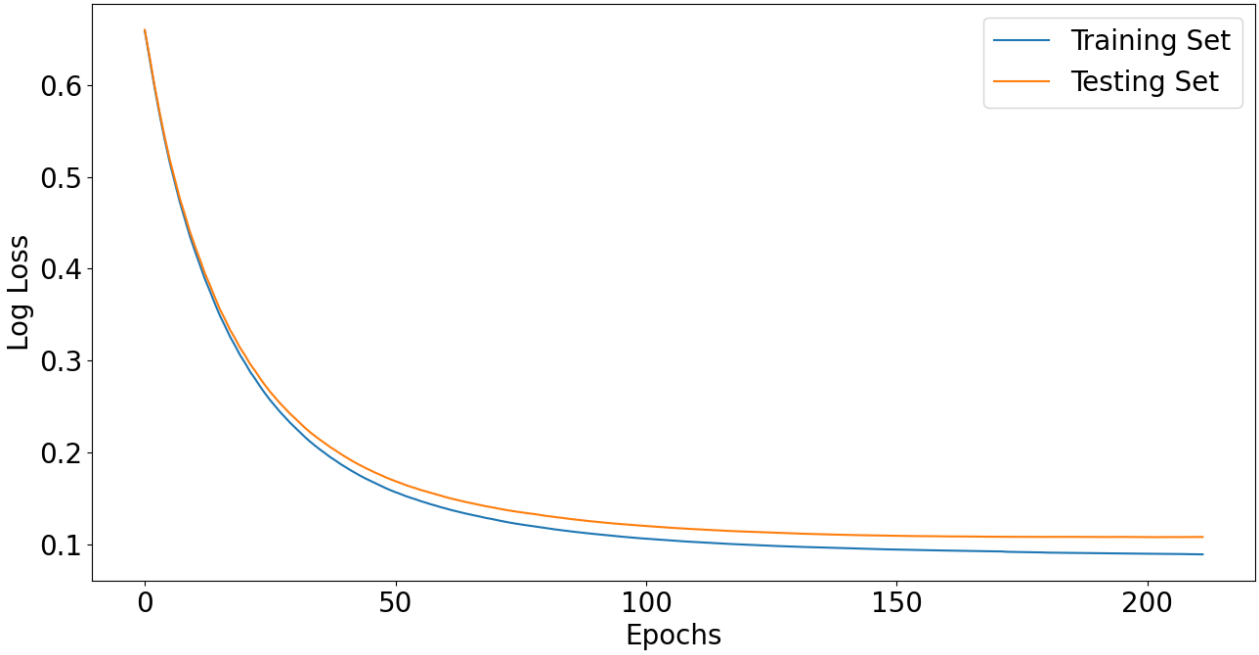


Figure 4.2: Plot for XGBoost Log Loss function

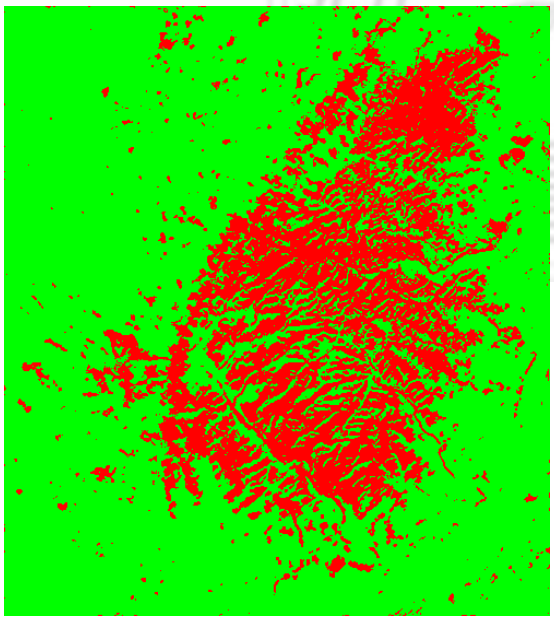
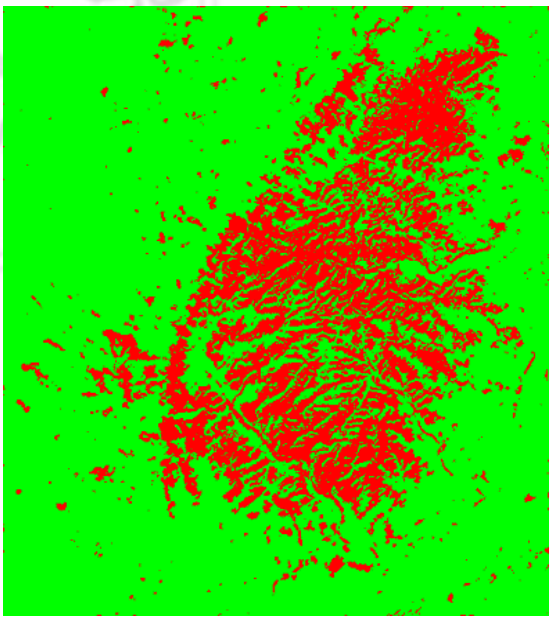
Table 3: Optimal Parameter values for XGBoost

Colsample_bytree	0.5054
Gamma	0.01
Learning Rate	0.04799
Max Depth	2
Min Child Weight	0.7646
Reg_Alpha	0.4921
Reg_Lambda	0.5066
Subsample	0.6831
No. of Estimators	203

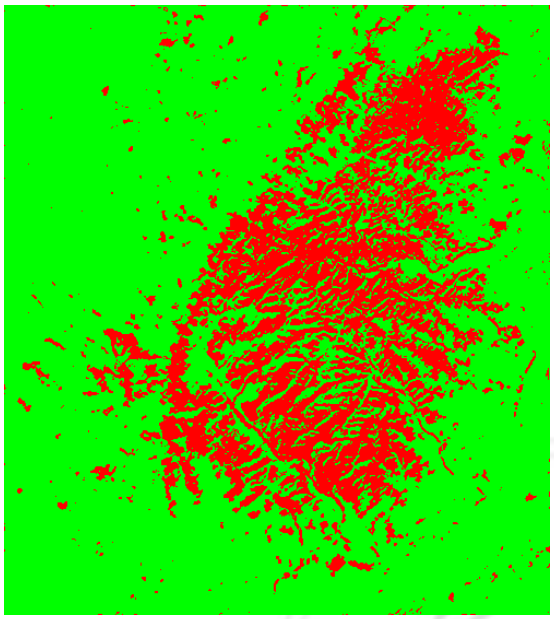
Table 4: Optimal Parameter values for SVM

Kernel	C	Gamma	Tolerance
RBF	1.356	148	1

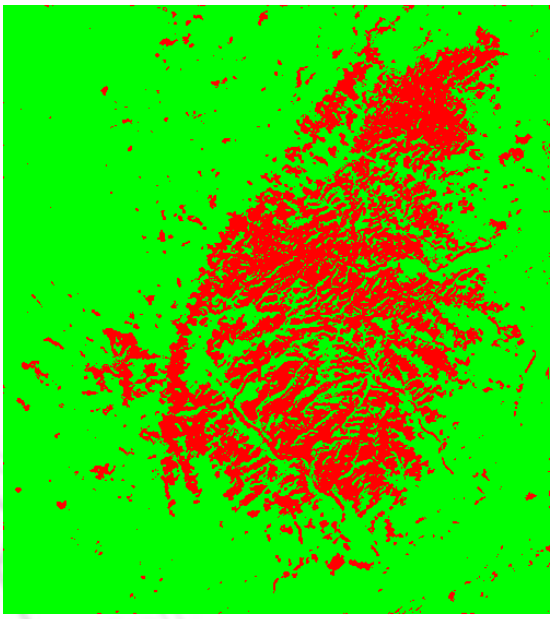
The trained models based on the benchmarks as outlined earlier were saved in the local system using the "pickle" open-source library for python. The saved model was then directly deployed on the entire normalized differenced image with stacked input features (NDVI and TBI) to detect landslides. The resulting landslide maps from all five supervised learning models with landslide predictions are depicted in Figure 4.3. The differenced image contained a total of 279,000 pixels on which binary classification was conducted i.e., landslide (red) and non-landslide (green). The number of pixels and total area in hectares according to the classification results from five models is presented in Table 5.

(a)  (b) 

(c)



(d)



(e)

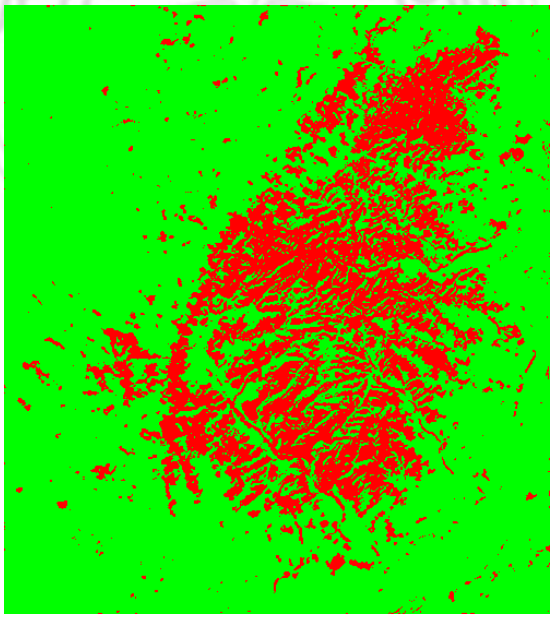


Figure 4.3: Landslide Inventory Maps from Supervised Models (a) SVM, (b) DT, (c) ET, (d) RF, (e) XGBoost

Table 5: Classification Results from Supervised Learning Algorithms

Algorithms	Landslide		Non-Landslide	
	No. of Pixels	Area (ha)	No. of Pixels	Area (ha)
SVM	69215	1081.48	209785	3277.89
DT	61574	962.094	217426	3397.28
ET	63723	995.672	215277	3363.70
RF	63601	993.766	215399	3365.61
XGBoost	63081	985.641	215919	3373.73

4.1.2 Unsupervised Approach

The topographic signatures were extracted based on 50 distinct clusters by applying K-means, Minibatch K-means, BIRCH, GMM clustering techniques. Following the clustering from the stacked features, a selection of clusters pertain to which group (landslide or non-landslide) was manually performed. Figure 4.4 demonstrates the Landslide Inventory Maps obtained following the manual designation of clusters into a landslide (red) and non-landslide region (green). The number of classified pixels and area (ha) for landslide and non- landslide regions are given in Table 6.

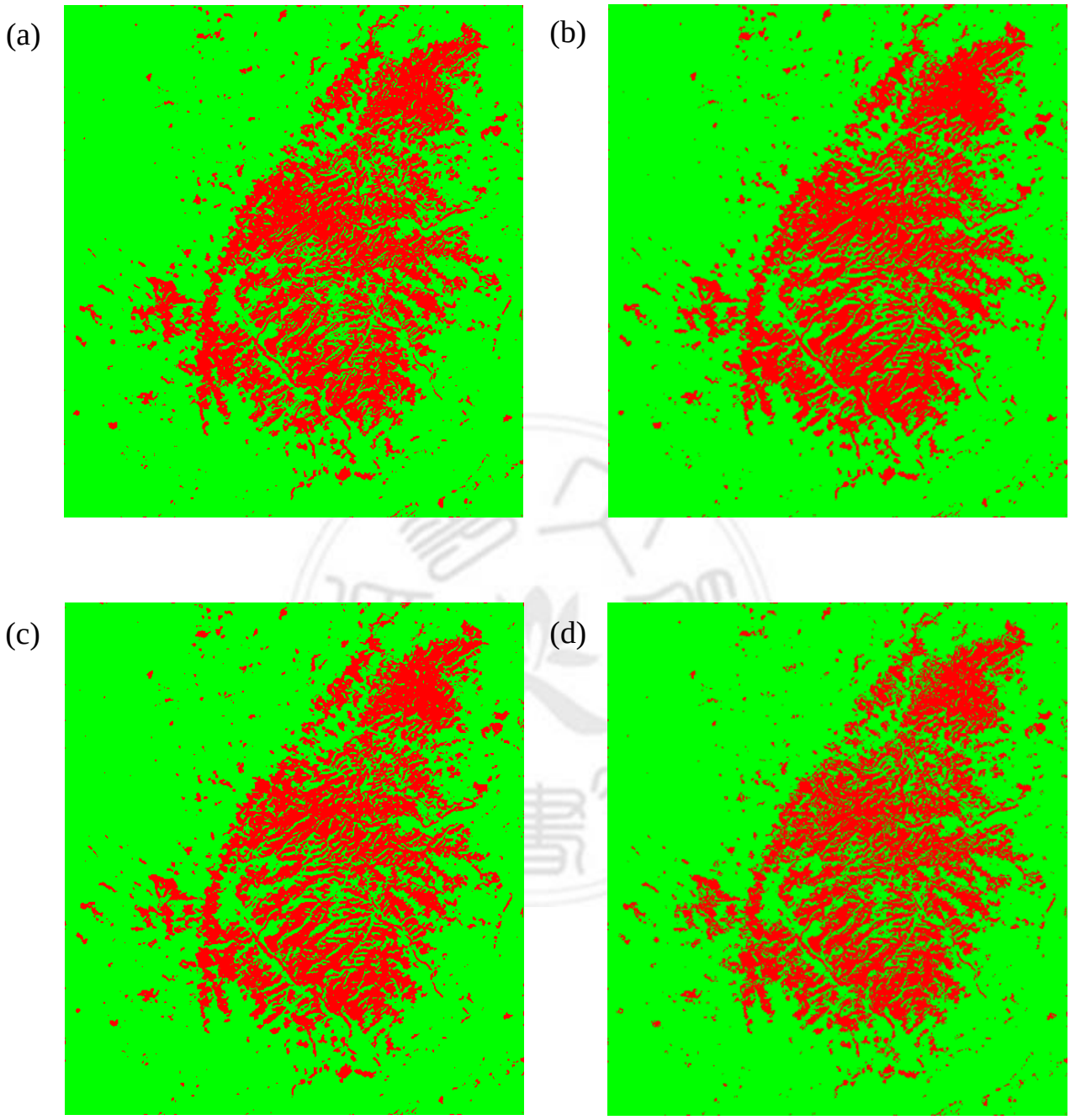


Figure 4.4: Landslide Inventory Maps from Unsupervised Clustering (a) K-means, (b) Minibatch K-means, (c) BIRCH, (d) GMM

Table 6: Classification Results from Unsupervised Learning Algorithms

Algorithms	Landslide		Non-Landslide	
	No. of Pixels	Area (ha)	No. of Pixels	Area (ha)
K-means	64808	1012.63	214192	3346.75
Minibatch K-means	67788	1059.19	211212	3300.19
BIRCH	65468	1022.94	213532	3336.44
GMM	61039	953.734	217961	3405.64

4.2 Accuracy Assessment and Comparison

First, the dataset extracted from the differenced image (stacked with NDVI and TBI) was split into a training set (25%) and testing set (75%) on which the supervised models were trained and evaluated for their fit. The trained models were then directly employed on the entire image for classification. The unsupervised algorithms were directly utilized on the entire differenced image for clustering followed by manual recoding of these clusters into landslide and non-landslide regions as discussed in the earlier sections. In this study, nine landslide maps were generated to investigate the capabilities of the proposed algorithms. For this purpose, a validation set was adopted with valid ground truth information of the study area. As outlined in section 3.6, a total of 52 polygons of which 40 polygons (1247 pixels) and 12 polygons (1436 pixels) that represent landslide and non-landslide terrains, respectively. The assessment of these maps was based on the criterion of User's Accuracy (UA), Producer's Accuracy (PA), Kappa Statistics (K), and Overall Accuracy (OA). Table 7 shows the quantitative statistics from error matrices derived for the generated landslide maps.

Table 7: Error Matrix for Inventory Maps based on Validation Set

Algorithm	Class Name	Pixels from the Landslide Maps		Row Total	UA (%)	PA (%)	OA (%)	K
		Non-Landslide	Landslide					
BIRCH	Non-Landslide	1287	3	1290	99.77	89.62	94.33	0.887
	Landslide	149	1244	1393	89.30	99.76		
GMM	Non-Landslide	1322	56	1378	95.94	92.06	93.66	0.873
	Landslide	114	1191	1305	91.26	95.51		
K-means	Non-Landslide	1343	16	1359	98.82	93.52	95.94	0.9187
	Landslide	93	1231	1324	92.98	98.72		
Minibatch K-means	Non-Landslide	1264	3	1267	99.76	88.02	93.48	0.8701
	Landslide	172	1244	1416	87.85	99.76		
SVM	Non-Landslide	1238	3	1241	99.76	86.21	92.51	0.8510
	Landslide	198	1244	1442	86.27	99.76		
DT	Non-Landslide	1315	3	1318	99.77	91.57	95.38	0.9077
	Landslide	121	1244	1365	91.14	99.76		
ET	Non-Landslide	1288	3	1291	99.77	89.69	94.37	0.8877
	Landslide	148	1244	1392	89.37	99.76		
RF	Non-Landslide	1316	4	1320	99.70	91.64	95.38	0.9077
	Landslide	120	1243	1363	91.20	99.68		
XGBoost	Non-Landslide	1320	5	1325	99.62	91.92	95.49	0.9099
	Landslide	116	1242	1358	91.46	99.60		

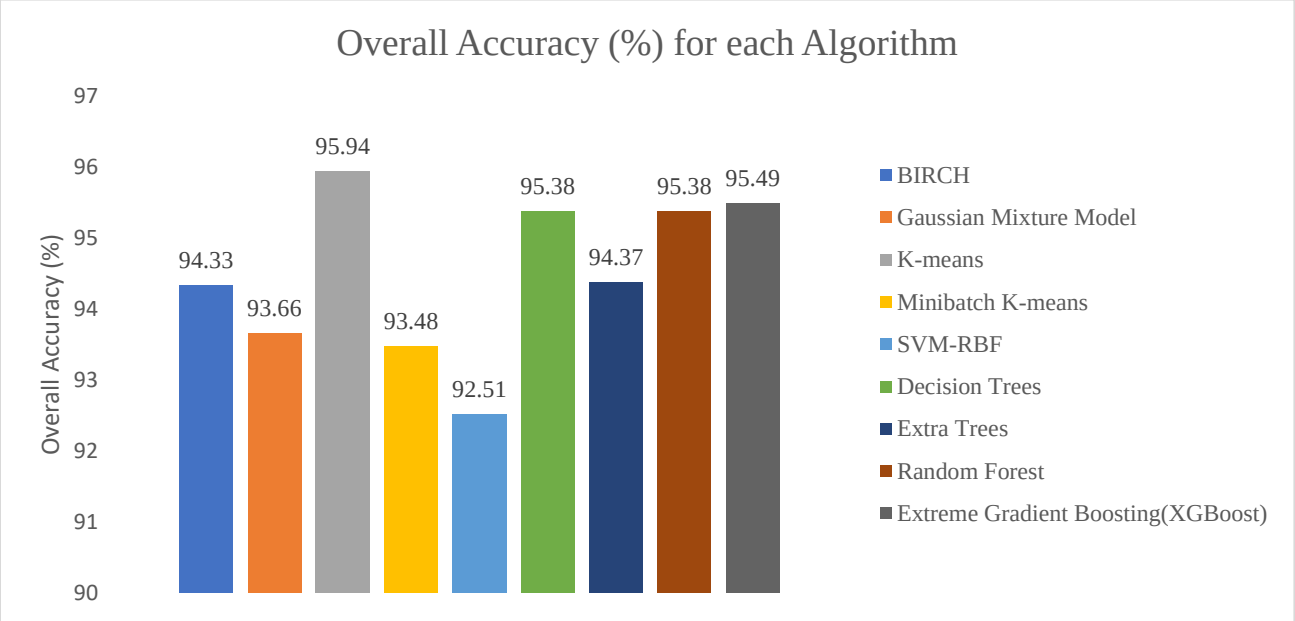


Figure 4.5: Bar chart for the Overall Accuracy from various Algorithms

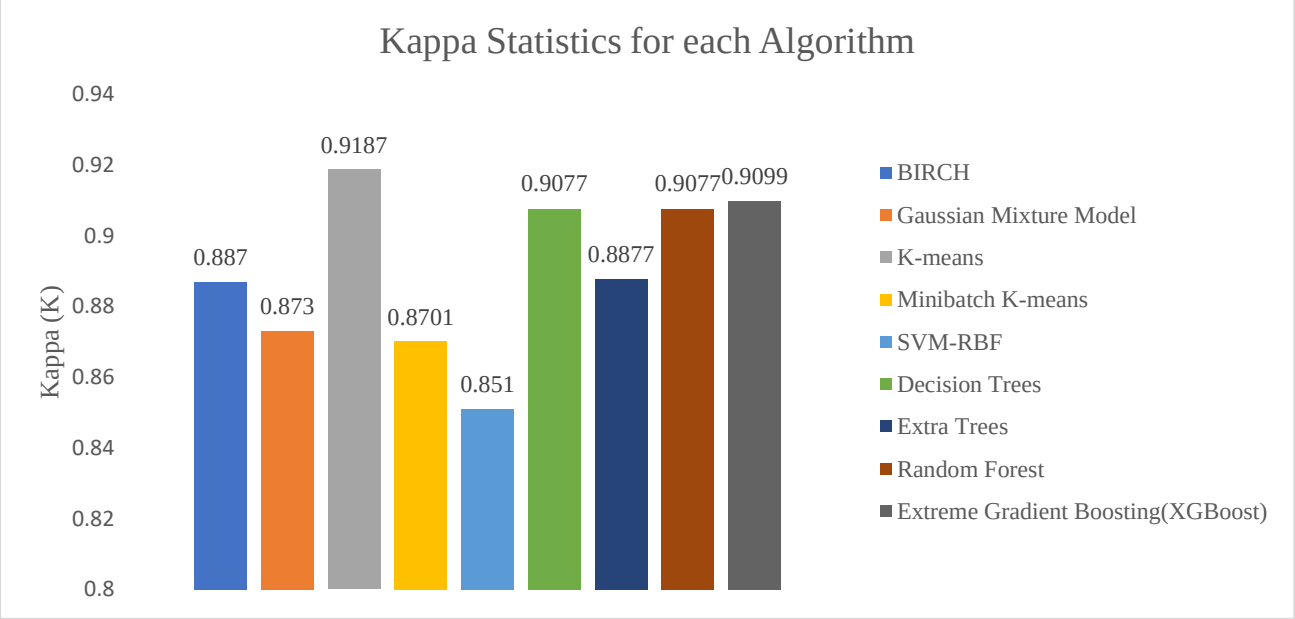


Figure 4.6: Bar chart for the Kappa Statistics from various Algorithms

The best K and OA (%) value was achieved by the K-means algorithm of 0.9187 and 95.94% closely followed by the XGBoost algorithm with 0.9099 and 95.49%, respectively. The lowest performance was achieved by the SVM model and Minibatch K-means for individual supervised and unsupervised approach, respectively. The lowest accuracy among all the algorithms was achieved by the SVM model with K and OA values at 0.8510 and 92.51%, respectively. The UA (%) for the non-landslide validation set was nearly identical across most of the algorithms with the lowest being in the GMM model of 95.94%, as for the landslide validation lowest UA resulted in the SVM model of 86.27%. The K-means algorithm achieved the highest UA for landslide (92.98%), while it was almost identical across XGBoost, GMM, DT, and RF. The remaining ET, Minibatch K-means, and BIRCH achieved UA just below 90%. On the other hand, PA in BIRCH, Minibatch K-means, SVM, DT, ET was the same at 99.76% on landslides, while the RF and XGBboost models were not far behind with 99.68% and 99.60%, respectively, and K-means yielded 98.72%. The lowest PA of 95.51% for landslides was observed in GMM. The highest PA for the non-landslide set was observed in the K-means of about 93.52% with the second being GMM and then XGBoost at 92.06% and 91.92%, respectively. The lowest PA for the non-landslide set was found in the SVM model of 86.21%. Figures 4.5 and 4.6 illustrate the bar plot for Kappa Statistics (K) and Overall Accuracy (OA) from all the proposed algorithms.

Chapter 5 Discussion & Conclusion

Landslide detection still faces several complexities owing to the various disparity in quality and spatial resolution of remotely sensed data. Recent research works aspire to implement the applicability of machine learning (data mining) to improve landslide modeling and mapping. This study was carried out in a densely vegetated mountainous region of Mt Jou-Jou in Central Taiwan. For this, four unsupervised and five supervised learning algorithms were utilized that were combined with pixel-based image differencing technique from multi-date SPOT-2 images to generate Landslide Inventory Maps. The use of remote sensing data eliminates the requirement of time-consuming and costly field surveys. The topographic terrains of landslides and non-landslide regions were recognized and the impact of two inversely related spectral indices (NDVI and TBI) were analyzed together in the same data frame on the landslide locations. Various training strategies greatly influence the results of supervised learning algorithms. During the training process, three benchmark metrics (Test Accuracy, Cross-Validation, RMSE) were monitored to eliminate overfitting or underfitting of the models. Some detailed parameter tuning was conducted on each supervised algorithm by considering a greater number of parameters to make the models more robust and accurate. The unsupervised approach only included separation of the whole differenced image into 50 distinct clusters and then designating each cluster into landslide and non-landslide groups.

Nine Landslide Inventory Maps were generated and comprehensively compared for their landslide recognition. A validation set that accurately depicts the on-site information was used to perform the quantitative analysis on these maps. The analysis implied that all the algorithms yielded good results when using input features (NDVI and TBI). However, K-means, XGBoost, RF, and DT algorithms

exhibited the highest rank in Overall Accuracies and Kappa Statistics. The results from K-means showed the highest classification accuracy, which is a rapid and time-effective technique. The second-best method was found to be XGBoost for landslide detection whereas DT and RF showed similar classification accuracy not lagging far behind. All the proposed algorithms achieved accuracies greater than 90%. The SVM model showed the lowest accuracy but yielded very good results. As illustrated in Table 7, the tradeoff between landslide and non-landslide validation pixels greatly impacted the overall assessments. For instance, algorithms such as BIRCH, Minibatch K-means, SVM, and ET yielded the highest accuracy for landslide validation set but at the cost of non-landslide inaccuracies. GMM achieved a higher number of correctly classified pixels for non-landslide but, the number of correct pixels in landslide greatly reduced. However, K-means, DT, RF, and XGBoost had significantly lower tradeoffs as a result achieved better overall accuracy. A common hypothesis has been adopted that most of the supervised algorithms will always yield better results, this may not imply that they will automatically outperform unsupervised techniques in all aspects. Our analysis revealed that unsupervised methods can still produce slightly better results in some ideal cases. Ghorbanzadeh (2019) gave similar implications that CNN outperforms SVM and RF on a case-by-case basis depending on its window and patch size.

The results of the comparison indicated that the inverse relation between NDVI and TBI quite agrees with the landslide signatures. In contrast, the NDVI was negative and TBI was positively correlated with landslides. Using such a dataset with simple attributes simplifies decision boundaries with fewer efforts on parameter optimization, reduces computational complexity, and improves the model. However, partitioning of landslide and non-landslide regions in densely vegetated terrains like Mt Jou-Jou is quite challenging and complex. Pradhan & Mezaal (2018) indicated

that accuracy may reduce even if the same algorithms or techniques are applied to other areas. The reasons could be different characteristics, sensor, environmental, and illumination conditions. Although, the present study can be used as a basis during the selection process when rapid and efficient results are of the essence. In this context, landslide identification techniques based on data mining are suitable to generate well-organized Landslide Inventory Maps. Various techniques have been proposed for this purpose. For instance, Chen et. al. (2014) used LiDAR-based DTM in conjunction with RF for landslide detection but, the best overall accuracy achieved was 78%. Tavakkoli et. al. (2019) integrated UAV and DEM data with a stacked machine learning model which included three models as the base classifier. The quantitative analysis showed that the overall accuracy of their study was 89.8%. Tran (2019) proposed a comparative study along with analysis on the effects of the predefined number of clusters on GMM and K-means in which both algorithms achieved nearly 87% overall accuracy at 2 to 4 clusters.

An additional investigation shall be conducted for the selection of appropriate feature extractors to generate Landslide Inventory Maps. As illustrated in Figures 4.3 and 4.4, all algorithms misclassified some of the dried streams and bare soil areas as landslides. Pradhan & Mezaal (2018) emphasized taking necessary measures to avoid misclassification of landcover classes more similar to landslides such as bare soil, man-made cut slope, etc. For instance, slope, terrain, texture, and other morphological characteristics may differ after landslides. Therefore, information derived from various sources such as topographic, hydrological, vegetation, lithology, and soil maps can be integrated to define a clearer boundary between similar land cover classes. However, this may not be true for all the wide variety of available data. Ghorbanzadeh et. al. (2019) effectively detected landslides by only integrating the spectral information with NDVI. Further addition of topographic

features slightly reduced the overall accuracy of the CNN model, but it was quite helpful for classification between settlement area and the landslide area that had similar spectral behavior. Hence, appropriate input feature analysis is critical for landslide assessment.

A few of the drawbacks in this study were: (1) limited feature analysis, and (2) no comparison with other popular deep learning techniques such as ANN, DNN, CNN, RNN, etc. (3) the models will fail to detect some of the deposited or displaced areas with no harm to their vegetation condition. In this study bitemporal images were adopted for detecting landslides, which were acquired just before and after the earthquake. Perhaps, the proposed unsupervised and supervised learning algorithms produced highly accurate landslide identification from this analysis since the differenced image from the bi-temporal images majorly included landslides and other change information such as land-use, land-cover, and water level change, etc., were hardly present. However, if the bitemporal images included change information, for example, new man-made structures such as roads, buildings, etc., it would have been difficult to classify these man-made structures from landslides with only spectral vegetation or brightness indices on the accounts of their identical structural features resulting in further deterioration of overall accuracy. Hence, the performance of the proposed algorithms in this study is highly dependent on the bi-temporal images (Lei et al., 2018) as well as the spectral nature. Overall, all the nine algorithms exhibited excellent performance for landslide assessment, but comparatively, the K-means algorithm yielded the best performance. Therefore, K-means is proposed as a promising technique for detecting landslides induced by earthquakes in mountainous regions similar to Mt Jou-Jou for its efficient and speedy production of landslide maps if integrated with SPOT-2 image. However, if a detailed analysis on landslide along with lower misclassification between similar

land cover classes is to be conducted then additional features derived from DEM such as topography, morphology, lithology, etc. that require additional processing and would not generate landslide maps rapidly but could provide in-depth information on landslides.

This study provides a comprehensive comparison of nine landslide inventory maps produced from four unsupervised and five supervised data mining algorithms was conducted to determine their further feasibility. The adopted system included: (1) a pixel-based image differencing technique, (2) suitable input features (NDVI & TBI), and (3) intelligent self-reliant decision-making techniques (data mining). The landslide maps were validated using reference pixels with accurate in-situ information. Error matrix for each map was computed based upon these pixels. Quantitative statistical analysis was based on "Producer's Accuracy" (PA), "User's Accuracy" (UA), "Overall Accuracy" (OA), and "Kappa" (K) which were computed to provide a baseline idea of the precision.

All the algorithms achieved accuracies higher than 90% maintaining their applicability for future studies. K-means algorithm achieved the highest Kappa (Overall Accuracy) of 0.9187 (95.94%) with a close follow-up from the XGBoost algorithm i.e., 0.9099 (95.49%). The Extra Trees and Decision Tree achieved similar accuracy of 0.9077 (95.38%). The lowest Kappa (Overall Accuracy) among all algorithms was observed in Support Vector Machines (SVM) i.e., 0.851 (92.51%). Recent research trends follow the adoption of newer deep learning algorithms with a time-consuming training process that heavily requires monitoring at each step. These research works occasionally ignored to compare their results with some of the simpler and older algorithms such as K-means, Decision Tree, etc. This study proves that in some exceptional cases some simpler models can outmatch or provide appreciable results in comparison to newer and complex data mining techniques.

Further contributions of this research towards the field of data mining-aided landslide detection are mentioned below:

(1) Establishment of a suitable non-complex dataset composed of inversely related spectral indices that complement each other for landslide detection.
(2) Detailed accuracy assessment with various quantitative statistical tools.
(3) Comparison between a wide range of data mining algorithms.
(4) Proposed a rapid landslide assessment system with reliable landslide inventory maps using K-means which is independent of human supervision.

Nonetheless, this study still requires further improvement and sophisticated statistical assessment to provide precise relevance in landslide analysis. Future research work could append input features into the data frame such as topographic, hydrological, geological data and analyze their responsiveness on the performance of these algorithms. The images could also be analyzed using the OBIA techniques. These algorithms can be further compared with some of the widely proclaimed deep learning models in the data science community such as ANN, CNN, DNN, and RNN, etc. As the validation and comparison of these inventory maps were based on a small patch of reference pixels, the future study would also include further comparison with inventory maps authorized by government agencies to accurately delineate the boundaries. In addition to this, the site can be further monitored and a Landslide Susceptibility Map can be produced to provide information on its current recovery status and vulnerable slopes using recent high-resolution satellite imagery from the SPOT-6 sensor with a pixel size of 6 x 6 (m). Although, such satellites contain very large datasets that K-means and GMM would fail to analyze. In such cases, Minibatch K-means and BIRCH would be an ideal choice for clustering as the algorithm initially splits the dataset into smaller subsamples (mini-batches), and the latter compacts & summarizes a large dataset while retaining as much information as possible followed by centroid selection and clustering. Nonetheless, the results

obtained from this research give implication towards a rapid and quick emergency response system by integrating SPOT-2 remote sensing data with K-means along with NDVI and TBI for Landslide Inventory Mapping.

References

1. Alkhasawneh, M. S., Ngah, U. K., Tay, L. T., Mat Isa, N. A., & Al-Batah, M. S. (2014). Modeling and testing landslide hazard using decision tree. Journal of Applied Mathematics, 2014.
2. Ari, Ç., & Aksoy, S. (2010, July). Unsupervised classification of remotely sensed images using Gaussian mixture models and particle swarm optimization. In 2010 IEEE International Geoscience and Remote Sensing Symposium (pp. 1859-1862). IEEE.
3. Chen, W., Li, X., Wang, Y., Chen, G., & Liu, S. (2014). Forested landslide detection using LiDAR data and the random forest algorithm: A case study of the Three Gorges, China. Remote sensing of environment, 152, 291-301.
4. Cintia Ganesha Putri, D., Leu, J. S., & Seda, P. (2020). Design of an Unsupervised Machine Learning-Based Movie Recommender System. Symmetry, 12(2), 185.
5. Danneels, G., Pirard, E., & Havenith, H. B. (2007, July). Automatic landslide detection from remote sensing images using supervised classification methods. In 2007 IEEE International Geoscience and Remote Sensing Symposium (pp. 3014-3017). IEEE.
6. Decision Classification in Python. Available Online: https://www.datacamp.com/community/tutorials/decision-tree-classification-python
7. Ghorbanzadeh, O., Blaschke, T., Gholamnia, K., Meena, S. R., Tiede, D., & Aryal, J. (2019). Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection. Remote Sensing, 11(2), 196.

8. Hardy, A., Ettritch, G., Cross, D. E., Bunting, P., Liywalii, F., Sakala, J., ... & Thomas, C. J. (2019). Automatic detection of open and vegetated water bodies using Sentinel 1 to map African malaria vector mosquito breeding habitats. Remote Sensing, 11(5), 593.
9. Hsu, C. C., & Lin, C. W. (2017). Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. IEEE Transactions on Multimedia, 20(2), 421-429.
10. Keyport, R. N., Oommen, T., Martha, T. R., Sajinkumar, K. S., & Gierke, J. S. (2018). A comparative analysis of pixel-and object-based detection of landslides from very high-resolution images. International journal of applied earth observation and geoinformation, 64, 1-11.
11. Lei, T., Xue, D., Lv, Z., Li, S., Zhang, Y., & K Nandi, A. (2018). Unsupervised change detection using fast fuzzy clustering for landslide mapping from very high-resolution images. Remote Sensing, 10(9), 1381.
12. Li, W., Prasad, S., & Fowler, J. E. (2013). Hyperspectral image classification using Gaussian mixture models and Markov random fields. IEEE Geoscience and Remote Sensing Letters, 11(1), 153-157.
13. Li, X., Cheng, X., Chen, W., Chen, G., & Liu, S. (2015). Identification of forested landslides using LiDar data, object-based image analysis, and machine learning algorithms. Remote sensing, 7(8), 9705-9726.
14. Lin, C. Y., Lo, H. M., Chou, W. C., & Lin, W. T. (2004). Vegetation recovery assessment at the Jou-Jou Mountain landslide area caused by the 921 Earthquake in Central Taiwan. Ecological Modelling, 176(1-2), 75-81.
15. Lin, W. T., Lin, C. Y., & Chou, W. C. (2006). Assessment of vegetation recovery and soil erosion at landslides caused by a catastrophic earthquake: a case study in Central Taiwan. Ecological Engineering, 28(1), 79-89.

16. Lin, W. T., Lin, C. Y., Tsai, J. S., & Huang, P. H. (2008). Eco-environmental changes assessment at the Chiufenershan landslide area caused by catastrophic earthquake in Central Taiwan. Ecological Engineering, 33(3-4), 220-232.
17. Ma, Z., Mei, G., & Piccialli, F. (2020). Machine learning for landslides prevention: a survey. Neural Computing and Applications, 1-27.
18. Nhu, V. H., Mohammadi, A., Shahabi, H., Ahmad, B. B., Al-Ansari, N., Shirzadi, A., ... & Chen, W. (2020). Landslide Detection and Susceptibility Modeling on Cameron Highlands (Malaysia): A Comparison between Random Forest, Logistic Regression and Logistic Model Tree Algorithms. Forests, 11(8), 830.
19. Park, S. J., Lee, C. W., Lee, S., & Lee, M. J. (2018). Landslide susceptibility mapping and comparison using decision tree models: A Case Study of Jumunjin Area, Korea. Remote Sensing, 10(10), 1545.
20. Pradhan, B., & Mezaal, M. R. (2018). Data mining-aided automatic landslide detection using airborne laser scanning data in densely forested tropical areas. Korean Journal of Remote Sensing.
21. Pradhan, A. M. S., & Kim, Y. T. (2020). Rainfall-Induced Shallow Landslide Susceptibility Mapping at Two Adjacent Catchments Using Advanced Machine Learning Algorithms. ISPRS International Journal of Geo-Information, 9(10), 569.
22. Ramos-Bernal, R. N., Vázquez-Jiménez, R., Romero-Calcerrada, R., Arrogante-Funes, P., & Novillo, C. J. (2018). Evaluation of unsupervised change detection methods applied to landslide inventory mapping using ASTER imagery. Remote Sensing, 10(12), 1987.

23. Reddy, A. C., Saraschandrika, A., & Reddy, A. V. (2020). Study of the Clustering Algorithms for Hyper Spectral Remote Sensing Images. Journal of Hyperspectral Remote Sensing v, 10(2), 117-121.
24. Ren, J., Wang, R., Liu, G., Feng, R., Wang, Y., & Wu, W. (2020). Partitioned Relief-F Method for Dimensionality Reduction of Hyperspectral Images. Remote Sensing, 12(7), 1104.
25. Sahin, E. K. (2020). Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest. SN Applied Sciences, 2(7), 1-17.
26. Sameen, M. I., & Pradhan, B. (2019). Landslide detection using residual networks and the fusion of spectral and topographic information. IEEE Access, 7, 114363-114373.
27. Sathiaraj, D., Huang, X., & Chen, J. (2019). Predicting climate types for the Continental United States using unsupervised clustering techniques. Environmetrics, 30(4), e2524.
28. Tan, K., Zhang, Y., Wang, X., & Chen, Y. (2019). Object-based change detection using multiple classifiers and multi-scale uncertainty analysis. Remote Sensing, 11(3), 359.
29. Tavakkoli Piralilou, S., Shahabi, H., Jarihani, B., Ghorbanzadeh, O., Blaschke, T., Gholamnia, K., ... & Aryal, J. (2019). Landslide detection using multi-scale image segmentation and different machine learning models in the higher himalayas. Remote Sensing, 11(21), 2575.
30. Tran, C. J., Mora, O. E., Fayne, J. V., & Lenzano, M. G. (2019). Unsupervised Classification for Landslide Detection from Airborne Laser Scanning. Geosciences, 9(5), 221.

31. Tsai, F., Hwang, J. H., Chen, L. C., & Lin, T. H. (2010). Post-disaster assessment of landslides in southern Taiwan after 2009 Typhoon Morakot using remote sensing and spatial analysis. Natural Hazards and Earth System Sciences, 10(10), 2179.
32. Understanding the Log loss function of XGBoost. Available Online: https://medium.com/datadriveninvestor/understanding-the-log-loss-function-of-xgboost-8842e99d975d
33. Vergunst, R. A Comparison of Clustering Algorithms. Final Year Project, University of Amsterdam, Amsterdam, Netherlands, July 13, 2017
34. Wan, S., Lei, T., & Chou, T. (2010). A novel data mining technique of analysis and classification for landslide problems. Natural hazards, 52(1), 211.
35. Wang, H., Zhang, L., Yin, K., Luo, H., & Li, J. (2020). Landslide identification using machine learning. Geoscience Frontiers, 12(1), 351-364.
36. Yang, M. D., Chen, S. C., & Tsai, H. P. (2017). A long-term vegetation recovery estimation for Mt. Jou-Jou using multi-date SPOT 1, 2, and 4 images. Remote Sensing, 9(9), 893.
37. Zhai, X., Liu, W., Yin, C., Peng, Y., Zhao, Y., Yang, Y., ... & Zhao, T. (2020). Automatic Unsupervised Landslide Detection Method Based on Single High-resolution Optical Image for Emergency Response. Sensors and Materials, 32(11), 4019-4036.

Appendix

Appendix I: Supervised Algorithms

(a) Support Vector Machines (SVM)

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.preprocessing import Normalizer
4. from sklearn.svm import SVC
5. from sklearn.model_selection import train_test_split, cross_val_score,
   GridSearchCV
6. from sklearn import metrics
7. from sklearn.metrics import mean_squared_error
8. from imblearn.over_sampling import SMOTE
9. import pickle
10.
11. #Importing Dataset into the Pandas Dataframe
12. df = pd.read_csv("Filename.csv", index_col= [])
13. X = df.drop("Column Header", axis= "columns")
14. y = df["class"]
15.
16. #Split into Training and Testing Set
17. X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=
18. 50, train_size= 0.25)
19.
20. #OverSampling
21. over = SMOTE(random_state= 50)
22. X_train_os, y_train_os = over.fit_sample(X_train, y_train)
23. X_test_os, y_test_os = over.fit_sample(X_test, y_test)
24.
25. #Normalization
26. mm = Normalizer()
27. X_train_mm = mm.fit_transform(X_train_os)
28. X_test_mm = mm.fit_transform(X_test_os)
29.
30. #Hyperparameter Tuning
31. svm = SVC(kernel= "rbf", random_state= 50)
32. parameters = {"C": [], "gamma": [], "tol": []}
33.
34. svm_h = GridSearchCV(svm, parameters, n_jobs= -1, verbose= True)
35. svm_h.fit(X_train_mm, y_train_os)
36.
37. print("The Best Score is: ", svm_h.best_score_)
38. print("Best Parameters are: ", svm_h.best_params_)
39. print(svm_h.best_estimator_)
40.
41. svm_tuned = svm_h.estimator
42. svm_tuned.fit(X_train_mm, y_train_os)
43. acc_tuned = svm_tuned.score(X_test_mm, y_test_os)
44. print("Accuracy with the Optimum Parameters: ", acc_tuned)
45.
46. #Training & Testing the Tuned Model
47. clf = SVC(kernel= "rbf", C= 1.356, gamma= 148, random_state= 50, tol=
```

```
48.  1)
49.  clf.fit(X_train_mm, y_train_os)
50.  accuracy = clf.score(X_test_mm, y_test_os)
51.  print("Test Accuracy: ", accuracy)
52.
53.  #Cross-Validation
54.  cv = cross_val_score(clf, X_train_mm, y_train_os, cv= 20)
55.  print(cv)
56.  print("CROSS-VALIDATION SCORE: ", cv.mean())
57.
58.  predict = clf.predict(X_test_mm)
59.  predict2 = clf.predict(X_train_mm)
60.
61.  #Root Mean Square Error for Testing Set
62.  mse = mean_squared_error(y_test_os, predict)
63.  rmse = np.sqrt(mse)
64.  print("Root Mean Square Error on Testing Set is: ", rmse)
65.
66.  #Root Mean Square Error for Training Set
67.  mse2 = mean_squared_error(y_train_os, predict2)
68.  rmse2 = np.sqrt(mse2)
69.  print("Root Mean Square Error on Training Set is: ", rmse2)
70.
71.  #Saving the Model
72.  pickle.dump(clf, open("Model Name.pickle", "wb"))
```

(b)Decision Tree (DT)

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.tree import DecisionTreeClassifier
4. from sklearn.model_selection import train_test_split, cross_val_score,
   GridSearchCV
5. from sklearn.preprocessing import Normalizer
6. from sklearn import metrics
7. from sklearn.metrics import mean_squared_error
8. from imblearn.over_sampling import SMOTE
9. import pickle
10.
11.  #Importing Dataset into the Pandas Dataframe
12.  df = pd.read_csv("Filename.csv", index_col= [])
13.  X = df.drop("Column Header", axis= "columns")
14.  y = df["class"]
15.
16.  #Split into Training and Testing Set
17.  X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=
18.  50, train_size= 0.25)
19.
20.  #Oversampling
21.  os = SMOTE(random_state= 50)
22.  X_train_os, y_train_os = os.fit_sample(X_train, y_train)
23.  X_test_os, y_test_os = os.fit_sample(X_test, y_test)
24.
25.  #Normalization
26.  mm = Normalizer()
```

```
X_train_mm = mm.fit_transform(X_train_os)
X_test_mm = mm.fit_transform(X_test_os)

#Hyperparameter Tuning
clf = DecisionTreeClassifier(random_state= 50)
params = {"max_depth": [], "criterion": [], "splitter": [],
"min_samples_split": [], "max_leaf_nodes": [], "min_samples_leaf": []}

clf_h = GridSearchCV(clf, params, n_jobs= -1, verbose= True)
clf_h.fit(X_train_mm, y_train_os)

print("Best Accuarcy: ", clf_h.best_score_)
print("Best Parameter: ", clf_h.best_params_)
print(clf_h.best_estimator_)

clf_tuned = clf_h.estimator
clf_tuned.fit(X_train_mm, y_train_os)
accuracy = clf_tuned.score(X_test_mm, y_test_os)
print("Accuracy with the Optimum Parameters: ", accuracy)

#Training & Testing the Model
clf = DecisionTreeClassifier(criterion= "entropy", max_depth= 19,
max_leaf_nodes= 39, min_samples_leaf= 3, min_samples_split= 2,
splitter= "best", random_state= 50)

clf.fit(X_train_mm, y_train_os)
accuracy = clf.score(X_test_mm, y_test_os)
print("Test Accuracy: ", accuracy)

#Cross-Validation
cv = cross_val_score(clf, X_train_mm, y_train_os, cv= 20)
print(cv)
print("CROSS-VALIDATION SCORE: ", cv.mean())

#Root Mean Square Error for Testing Set
predict = clf.predict(X_test_mm)
mse = mean_squared_error(y_test_os, predict)
rmse = np.sqrt(mse)
print("Root Mean Square Error on Testing Set is: ", rmse)

#Root Mean Square Error for Training Set
predict2 = clf.predict(X_train_mm)
mse2 = mean_squared_error(y_train_os, predict2)
rmse2 = np.sqrt(mse2)
print("Root Mean Square Error on Training Set is: ", rmse2)

#Saving the Model
pickle.dump(clf, open("Model Name.pickle", "wb")
```

(c) Extra Trees (ET)

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.ensemble import ExtraTreesClassifier
4. from sklearn.preprocessing import Normalizer
5. from sklearn.model_selection import cross_val_score, GridSearchCV,
   train_test_split
6. from sklearn.metrics import mean_squared_error
7. import pickle
8. from imblearn.over_sampling import SMOTE
9.
10. #Importing Dataset into the Pandas Dataframe
11. df = pd.read_csv("Filename.csv", index_col= [])
12. X = df.drop("Column Header", axis= "column")
13. y = df["class"]
14.
15. #Split into Training and Testing Set
16. X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=
17. 50, train_size= 0.25)
18.
19. #Oversampling
20. os = SMOTE(random_state= 50)
21. X_train_os, y_train_os = os.fit_sample(X_train, y_train)
22. X_test_os, y_test_os = os.fit_sample(X_test, y_test)
23.
24. #Normalization
25. mm = Normalizer()
26. X_train_mm = mm.fit_transform(X_train_os)
27. X_test_mm = mm.fit_transform(X_test_os)
28.
29. #Hyperparameter Tuning
30. clf = ExtraTreesClassifier(random_state= 50)
31. params = {"max_depth": [], "n_estimators": [], "criterion": [],
32. "min_samples_split": [], "max_leaf_nodes": [], "min_samples_leaf": []}
33.
34. clf_h = GridSearchCV(clf, params, n_jobs= -1, verbose= True)
35. clf_h.fit(X_train_mm, y_train_os)
36.
37. print("Best Accuarcy: ", clf_h.best_score_)
38. print("Best Parameter: ", clf_h.best_params_)
39. print(clf_h.best_estimator_)
40.
41. clf_tuned = clf_h.estimator
42. clf_tuned.fit(X_train_mm, y_train_os)
43. accuracy = clf_tuned.score(X_test_mm, y_test_os)
44. print("Accuracy with the Optimum Parameters: ", accuracy)
45.
46. #Training & Testing the Model
47. clf = ExtraTreesClassifier(criterion= "gini", max_depth= 17,
48. max_leaf_nodes= 45, min_samples_leaf= 2, min_samples_split= 2,
49. n_estimators= 84, random_state= 50)
50.
51. clf.fit(X_train_mm, y_train_os)
52. accuracy = clf.score(X_test_mm, y_test_os)
53. print("Test Accuracy: ", accuracy)
```

```
54. #Cross-Validation
55. cv = cross_val_score(clf, X_train_mm, y_train_os, cv= 20)
56. print(cv)
57. print("CROSS-VALIDATION SCORE: ", cv.mean())
58.
59. #Root Mean Square Error for Testing Set
60. predict = clf.predict(X_test_mm)
61. mse = mean_squared_error(y_test_os, predict)
62. rmse = np.sqrt(mse)
63. print("Root Mean Square Error on Testing Set is: ", rmse)
64.
65. #Root Mean Square Error for Training Set
66. predict2 = clf.predict(X_train_mm)
67. mse2 = mean_squared_error(y_train_os, predict2)
68. rmse2 = np.sqrt(mse2)
69. print("Root Mean Square Error on Training Set is: ", rmse2)
70.
71. #Saving the Model
72. pickle.dump(clf, open("Model Name.pickle", "wb"))
```

(d)Random Forest (RF)

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.ensemble import RandomForestClassifier
4. from sklearn.model_selection import train_test_split, cross_val_score,
   GridSearchCV
5. from sklearn.preprocessing import Normalizer
6. from sklearn import metrics
7. from sklearn.metrics import mean_squared_error
8. from imblearn.over_sampling import SMOTE
9. import pickle
10.
11. #Importing Dataset into the Pandas Dataframe
12. df = pd.read_csv("Filename.csv", index_col= [])
13. X = df.drop("Column Header", axis= "columns")
14. y = df["class"]
15.
16. #Split into Training and Testing Set
17. X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=
18. 50, train_size= 0.25)
19.
20. #Oversampling
21. os = SMOTE(random_state= 50)
22. X_train_os, y_train_os = os.fit_sample(X_train, y_train)
23. X_test_os, y_test_os = os.fit_sample(X_test, y_test)
24.
25. #Normalization
26. mm = Normalizer()
27. X_train_mm = mm.fit_transform(X_train_os)
28. X_test_mm = mm.fit_transform(X_test_os)
29.
30. #Hyperparameter Tuning
31. clf = RandomForestClassifier(random_state= 50)
32. params = {"max_depth": [], "n_estimators": [], "criterion": [],
```

```
"min_samples_split": [], "max_leaf_nodes": [], "min_samples_leaf": [],
"max_samples": []}

clf_h = GridSearchCV(clf, params, n_jobs= -1, verbose= True)
clf_h.fit(X_train_mm, y_train_os)

print("Best Accuarcy: ", clf_h.best_score_)
print("Best Parameter: ", clf_h.best_params_)
print(clf_h.best_estimator_)

clf_tuned = clf_h.estimator
clf_tuned.fit(X_train_mm, y_train_os)
accuracy = clf_tuned.score(X_test_mm, y_test_os)
print("Accuracy with the Optimum Parameters: ", accuracy)

#Training & Testing the Model
clf = RandomForestClassifier(criterion= "gini", max_depth= 7,
max_leaf_nodes= 36, max_samples= 0.9644, min_samples_leaf= 3,
min_samples_split= 7, n_estimators= 105, random_state= 50)

clf.fit(X_train_mm, y_train_os)
accuracy = clf.score(X_test_mm, y_test_os)
print("Test Accuracy: ", accuracy)

#Cross-Validation
cv = cross_val_score(clf, X_train_mm, y_train_os, cv= 20)
print(cv)
print("CROSS-VALIDATION SCORE: ", cv.mean())

#Root Mean Square Error for Testing Set
predict = clf.predict(X_test_mm)
mse = mean_squared_error(y_test_os, predict)
rmse = np.sqrt(mse)
print("Root Mean Square Error on Testing Set is: ", rmse)

#Root Mean Square Error for Training Set
predict2 = clf.predict(X_train_mm)
mse2 = mean_squared_error(y_train_os, predict2)
rmse2 = np.sqrt(mse2)
print("Root Mean Square Error on Training Set is: ", rmse2)

#Saving the Model
pickle.dump(clf, open("Model Name.pickle", "wb"))
```

(e) Extreme Gradient Boosting (XGBoost)

```
1. import numpy as np
2. import pandas as pd
3. from sklearn.preprocessing import Normalizer
4. from sklearn.model_selection import train_test_split, cross_val_score,
   GridSearchCV
5. from imblearn.over_sampling import SMOTE
6. from sklearn.metrics import mean_squared_error
7. import matplotlib.pyplot as plt
8. import xgboost as xgb
9. import pickle
10.
11. #Importing Dataset into the Pandas Dataframe
12. df = pd.read_csv("Filename.csv", index_col= [])
13. X = df.drop("Column Header", axis= "columns")
14. y = df["class"]
15.
16. #Split into Training and Testing Set
17. X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=
18. 50, train_size= 0.25)
19.
20. #Oversampling
21. os = SMOTE(random_state= 50)
22. X_train_os, y_train_os = os.fit_sample(X_train, y_train)
23. X_test_os, y_test_os = os.fit_sample(X_test, y_test)
24.
25. #Normalization
26. mm = Normalizer()
27. X_train_mm = mm.fit_transform(X_train_os)
28. X_test_mm = mm.fit_transform(X_test_os)
29.
30. #Hyperparameter Tuning
31. clf = XGBClassifier(booster= "gbtree", random_state= 50, n_estimators=
32. 50)
33. params = {"learning_rate": [], "min_child_weight": [], "max_depth":
34. [], "gamma": [], "subsample": [], "colsample_bytree": [], "reg_lambda":
35. [], "reg_alpha": []}
36.
37. clf_h = GridSearchCV(clf, params, n_jobs= -1, verbose= True)
38. clf_h.fit(X_train_mm, y_train_os)
39.
40. print("Best Accuarcy: ", clf_h.best_score_)
41. print("Best Parameter: ", clf_h.best_params_)
42. print(clf_h.best_estimator_)
43.
44. clf_tuned = clf_h.estimator
45. clf_tuned.fit(X_train_mm, y_train_os)
46. accuracy = clf_tuned.score(X_test_mm, y_test_os)
47. print("Accuracy with the Optimum Parameters: ", accuracy)
48.
49. #Estimating the Best Iteration (Epoch)
50. clf = xgb.XGBClassifier(booster= "gbtree", colsample_bytree= 0.5045,
51. gamma= 0.01, learning_rate= 0.04799, max_depth= 2, min_child_weight=
52. 0.7646, reg_alpha= 0.4921, reg_lambda= 0.5066, subsample= 0.6831,
53. n_estimators= 1000)
```

```
54. *Early Stopping Criterion
55. eval_set = [(X_train_mm, y_train_os),(X_test_mm, y_test_os)]
56. clf.fit(X_train_mm, y_train_os, eval_metric= ["error", "logloss"],
57. eval_set = eval_set, verbose= True, early_stopping_rounds= 10)
58.
59. accuracy = clf.score(X_test_mm, y_test_os)
60. print("Accuracy at the Termination Round: ", accuracy)
61. results = clf.evals_result()
62. epochs = len(results['validation_0’]['error'])
63. x_axis = range(0, epochs)
64. print("Best Iteration: ", clf.best_iteration)
65.
66. #Graph Plot for Classification Error
67. plt.figure(figsize= (16,9))
68. plt.plot(x_axis, results['validation_0']['error'], label= 'Training
69. Set')
70. plt.plot(x_axis, results['validation_1']['error'], label= 'Testing
71. Set')
72. plt.xticks(fontsize= 20)
73. plt.yticks(fontsize= 20)
74. plt.legend(fontsize= 20)
75. plt.ylabel('Classification Error', fontsize= 20)
76. plt.xlabel('Epochs', fontsize= 20)
77. plt.show()
78.
79. #Graph Plot for Log Loss Function
80. plt.figure(figsize= (16,9))
81. plt.plot(x_axis, results['validation_0']['logloss'], label= 'Training
82. Set')
83. plt.plot(x_axis, results['validation_1']['logloss'], label= 'Testing
84. Set')
85. plt.xticks(fontsize= 20)
86. plt.yticks(fontsize= 20)
87. plt.legend(fontsize= 20)
88. plt.ylabel('Log loss', fontsize= 20)
89. plt.xlabel('Epochs', fontsize= 20)
90. plt.show()
91.
92. #Training & Testing the Model with Best Iteration (Epoch)
93. clf = xgb.XGBClassifier(booster= "gbtree", colsample_bytree= 0.5045,
94. gamma= 0.01, learning_rate= 0.04799, max_depth= 2, min_child_weight=
95. 0.7646, reg_alpha= 0.4921, reg_lambda= 0.5066, subsample= 0.6831,
96. n_estimators= 203)
97.
98. clf.fit(X_train_mm, y_train_os)
99. accuracy = clf.score(X_test_mm, y_test_os)
100. print("Test Accuracy: ", accuracy)
101.
102. #Cross-Validation
103. cv = cross_val_score(clf, X_train_mm, y_train_os, cv= 20)
104. print(cv)
105. print("CROSS-VALIDATION SCORE: ", cv.mean())
106.
107. #Root Mean Square Error for Testing Set
108. predict = clf.predict(X_test_mm)
109. mse = mean_squared_error(y_test_os, predict)
110. rmse = np.sqrt(mse)
```

```
print("Root Mean Square Error on Testing Set is: ", rmse)

#Root Mean Square Error for Training Set
predict2 = clf.predict(X_train_mm)
mse2 = mean_squared_error(y_train_os, predict2)
rmse2 = np.sqrt(mse2)
print("Root Mean Square Error on Training Set is: ", rmse2)

#Saving the Model
pickle.dump(clf, open("Model Name.pickle", "wb"))
```

(f) Model Deployment

```
import pandas as pd
import pickle
from sklearn.preprocessing import Normalizer

#Normalizing the Entire Image
df = pd.read_csv("Image ASCII Filename.csv", index_col= [])

array = df.values  (Storing the values into a variable)
X = array[: :]     (Converting the values into an array)

 mm = Normalizer()
 mm.fit(X)
 X_mm = mm.transform(X)

 #Saving the Normalized Values into CSV Format
 X_mm_df = pd.DataFrame(X_mm, columns= ["Column Header"])
 X_mm_df.to_csv("Normalized Image ASCII Filename.csv", index= False)

 #Importing Normalized Image into the Pandas Dataframe
 df = pd.read_csv("Normalized Image ASCII Filename.csv", index_col= [])

 #Loading the Saved Model for Prediction
 model = pickle.load(open("Model Name.pickle", "rb"))
 predict = model.predict(df)

 #Saving the Classification Results
 predict_df = pd.DataFrame(predict, columns= ["Column Header"])
 predict_df.to_csv("Filename.csv", index= False)
```

Appendix II: Unsupervised Algorithms

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.cluster import Birch, MiniBatchKMeans, KMeans
4. from sklearn.mixture import GaussianMixture
5.
6. #Importing Normalized Image into the Pandas Dataframe
7. df = pd.read_csv("Normalized Image ASCII Filename.csv", index_col= [])
8.
9. #BIRCH Clustering
10.  cluster = Birch(threshold= 0.01, n_clusters= 50, branching_factor= 50)
11.  prediction = cluster.fit_predict(df)
12.
13.  pr_df = pd.DataFrame(prediction, columns= ["Column Header"])
14.  print(prediction)
15.  *Saving the Clustering Results
16.  pr_df.to_csv("Filename.csv", index= False)
17.
18.
19.  #Minibatch K-means Clustering
20.  cluster = MiniBatchKMeans(n_clusters= 50, init= "k-means++", max_iter=
21.  150, batch_size= 150, random_state= 10, tol= 0.001, max_no_improvement=
22.  10, reassignment_ratio= 0.1)
23.  prediction = cluster.fit_predict(df)
24.
25.  pr_df = pd.DataFrame(prediction, columns= ["Column Header"])
26.  print(prediction)
27.  *Saving the Clustering Results
28.  pr_df.to_csv("Filename.csv", index= False)
29.
30.
31.  #Gaussian Mixture Model
32.  cluster = GaussianMixture(n_components= 50, verbose= True)
33.  prediction = cluster.fit_predict(df)
34.
35.  pr_df = pd.DataFrame(prediction, columns= ["Column Header"])
36.  print(prediction)
37.  *Saving the Clustering Results
38.  pr_df.to_csv("Filename.csv", index= False)
39.
40.
41.  #K-means
42.  cluster = KMeans(n_clusters= 50, max_iter= 100, random_state= 0, init=
43.  "k-means++")
44.  prediction = cluster.fit_predict(df)
45.
46.  pr_df = pd.DataFrame(prediction, columns= ["Column Header"])
47.  print(prediction)
48.  *Saving the Clustering Results
49.  pr_df.to_csv("Filename.csv", index= False)
```