JADAVPUR UNIVERSITY

MASTER DEGREE THESIS

Prediction of Landslide Triggers in India using Stacked Ensemble Models

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Engineering

in the

Computer Science and Engineering

by
Tirtha Das
Class Roll No. 002210502012
Examination Roll No. M4CSE24009
Registration No. 163563 of 2022-2023

Under the Guidance of
Prof. Sarbani Roy
Department of Computer Science and Engineering
Jadavpur University
Kolkata-700032.

May 27, 2024

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

CERTIFICATE OF RECOMMENDATION

This is to certify that the work embodied in this thesis entitled **"Prediction of Landslide Triggers in India using Stacked Ensemble Models"** has been satisfactorily completed by **Tirtha Das** (Registration Number 163563 of 2022 - 2023; Class Roll No. 002210502012; Examination Roll No. M4CSE24009). It is a bonafide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata, for partial fulfillment of the requirements for the awarding of the **Master of Engineering** in **Computer Science and Engineering** degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2022 - 2024.

Prof. Sarbani Roy,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

Prof. Nandini Mukhopadhyay,
HOD, Department of Computer Science and Engineering,
Jadavpur University.

Prof. Dipak Laha ,
DEAN, Faculty of Engineering and Technology,
Jadavpur University.

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled **"Prediction of Landslide Triggers in India using Stacked Ensemble Models"** is a bonafide record of work carried out by **Tirtha Das** (Registration Number 163563 of 2022 - 2023; Class Roll No. 002210502012; Examination Roll No. M4CSE24009) in partial fulfillment of the requirements for the award of the degree of **Master of Engineering** in **Computer Science and Engineering** degree of the Department of Computer Science and Engineering, during the period of September 2022 to May 2024. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose of which it has been submitted.

_____
Signature of Examiner 1
Date:

_____
Signature of Examiner 2
Date:

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that the thesis entitled **"Prediction of Landslide Triggers in India using Stacked Ensemble Models"** contains literature survey and original research work by the undersigned candidate, as a part of her degree of **Master of Engineering** in **Computer Science and Engineering, Jadavpur University**. All the information have been obtained and presented in accordance with academic rules and ethical conduct.
I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Tirtha Das

Examination Roll No.: M4CSE24009

Registration No.: 163563 of 2022 - 2023

Thesis Title: Prediction of Landslide Triggers in India using Stacked Ensemble Models

Signature of the Candidate:

Dedicated to my family, friends and well-wishers
for their endless love, support and encouragement

Acknowledgements

The completion and success of my thesis were made possible by the invaluable guidance and support of numerous individuals. I am deeply grateful for their assistance throughout this project and wish to express my sincere appreciation for their contributions to my achievements.

I owe my deep gratitude to my supervisor and thesis guide **Prof. Sarbani Roy**, who took a keen interest in my thesis work and guided me all along till the completion of my thesis work by providing all the necessary information for developing a good end-to-end solution. I am extremely thankful to her for her assistance and dedicated involvement in every step of the process. Her constant suggestions and editorial enhancements have been an everlasting source of motivation and inspiration to me. It has been an honour to work under a person with wisdom yet a kind, simplistic attitude.

I had the pleasure of working in the Stars lab. I acknowledge the generous collaboration and assistance of every member of the Stars lab.

I would also like to thank **Prof. Nandini Mukhopadhyay**, Head of the department, Department of Computer Science and Engineering at Jadavpur University, for extending her help to complete this thesis.

In addition to my supervisor, I wish to express my heartfelt appreciation to **Mr. Asif Iqbal Middya**, research scholar at the Department of Computer Science and Engineering at Jadavpur University. His unwavering support, inspiration, invaluable insights, and research expertise were indispensable to me. Without his energy, encouragement, and unwavering optimism, unconditional support this thesis would not have progressed as far as it did.

Most significantly, none of this would have been possible without my family's & friend's constant support. It goes without saying that my mother **Mrs. Ratna Das** and my father **Mr. Samar Kumar Das** have always supported me when I was in need. This thesis would not have been possible if it weren't for their patience and unwavering support.

Last but not least, I would like to express my gratitude to all of the professors at Jadavpur University, as well as to my seniors, classmates, and well-wishers who have

served as a continual source of support and motivation for me as I worked on my thesis. I want to express my gratitude and appreciation to every one of them for helping to make this beautiful journey possible.

Tirtha Das

Registration No. 163563 of 2022-2023

Examination Roll No. M4CSE24009

Class Roll No. 002210502012

Dept. of Computer Science & Engineering

Jadavpur University

Kolkata, India

JADAVPUR UNIVERSITY

Abstract

Faculty of Engineering and Technology, Jadavpur University
Computer Science and Engineering

Master of Engineering

Prediction of Landslide Triggers in India using Stacked Ensemble Models

by Tirtha Das

Landslides are among the most devastating and recurrent natural disasters, impacting numerous mountainous regions worldwide. India is no exception, with frequent landslides affecting vital economic sectors such as transportation and agriculture, often resulting in significant loss of life. A landslide trigger plays a crucial role in initiating the downward movement of rock, earth or debris on a slope. Triggers include natural events like heavy rainfall, earthquakes and rapid snowmelt, as well as human activities like construction, deforestation and mining, which disrupt slope stability and cause significant damage and loss of life. Accurately identifying the triggers of landslides is essential for reducing risks and mitigating their impact. Many researchers have explored landslide prediction, but only a few have focused on landslide triggers. The main challenge with India's landslide dataset is its inherent imbalance. To address this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. SMOTE generates synthetic examples for the minority classes, thereby balancing the dataset. This approach improves the model's ability to correctly identify and predict landslides triggered by less frequent events, leading to more robust and reliable predictions across all classes. To predict landslide triggers, a stacked-based machine learning model has been developed, which outperformed other baseline models with an accuracy of 94.6% on unseen data. This achievement underscores the effectiveness of the stacked ensemble model in capturing the complex relationships between various factors contributing to landslide triggers. The robust performance of the stacked ensemble model highlights its potential as a valuable tool for improving landslide prediction and enhancing public safety.

Keywords: *Landslide trigger, Machine Learning, Multilayer Perceptron, Classification, Stacking, Smote*

Contents

Certificate of Recommendation i

Certificate of Approval ii

Declaration Of Originality and Compliance of Academic Ethics iii

Acknowledgements v

Abstract vii

1 Introduction 1
- 1.1 Problem Overview . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objective . . . 3
- 1.4 Contribution . . . 3
- 1.5 Organization . . . 4

2 Related works 5
- 2.1 Shallow machine learning models for landslide predictions . . . 5
- 2.2 Deep learning models of landslide predictions . . . 6
- 2.3 Hybrid machine learning models for landslide predictions . . . 7

3 Methodology 11
- 3.1 Problem Statement . . . 11
- 3.2 Study Area . . . 12
- 3.3 Data collection and description . . . 12
- 3.4 Data preprocessing . . . 14
- 3.5 Proposed model . . . 16
  - 3.5.1 Stacked machine learning model . . . 17

4 Result and Analysis 21
- 4.1 Experimental setup . . . 21
- 4.2 Baselines and Performance metrics . . . 21
- 4.3 Result and Discussion . . . 24

5 Conclusion and Future Work 31
- 5.1 Conclusion . . . 31
- 5.2 Future Work . . . 32

List of Figures

3.1 Locations in India where landslides have been occurred 13
3.2 Work flow diagram of the study 16
3.3 Architecture of proposed stacked ensemble learning model 18
3.4 Number of base model vs accuracy plot 19

4.1 AUC score of various meta learners 27
4.2 Confusion matrix for all meta learner in Stack ensemble learning . . . 29

List of Tables

2.1 Existing literature on landslide prediction 9

3.1 Features and descriptions 15
3.2 Specifications of Artificial Neural Network Models 18

4.1 Performance Comparison of the Proposed Model with Baseline ML Models 23
4.2 Baseline Models' Train and Test Times (in milliseconds) 24
4.3 Performance of Stack ensemble learning with various meta learner . . 27
4.4 Train and test time duration of different meta learner in mean $\pm$ sd format (sec) 28

List of Abbreviations

CSV	**C**omma **S**eparated **V**alues
VIF	**V**ariance **I**nflation **F**actor
NDVI	**N**ormalized **D**ifference **V**egetation **I**ndex
NDWI	**N**ormalized **D**ifference **W**ater **I**ndex
SPI	**S**tream **P**ower **I**ndex
SMOTE	**S**ynthetic **M**inority **O**versampling **T**echnique
AB	**A**da **B**oost
DT	**D**ecision **T**ree
GB	**G**radient **B**oosting
KNN	**K** **N**earest **N**eighbour
KSVM	**K**ernel **S**upport **V**ector **M**achine
LR	**L**ogistic **R**egression
NB	**N**aive **B**ayes
RF	**R**andom **F**orest
SVM	**S**upport **V**ector **M**achine
XGB	**XGB**oost
LGBM	**L**ight **G**radient**B**oosting **M**achine
ANN	**A**rtificial Neural Network
MLP	**M**ulti **L**ayer **P**erceptron
ML	**M**achine **L**earning
ROC	**R**eceiver **O**perating **C**haracteristic
AUC	**A**rea **U**nder the **C**urve
LC	**L**and **C**over
LST	**L**and **S**urface **T**emperature

Chapter 1

Introduction

1.1 Problem Overview

Landslides are classified as natural disasters, often instigated by a multitude of natural phenomena such as earthquakes, windstorms, heavy rainfall and various other factors collectively known as landslide triggers. These triggers initiate the downward movement of rock, earth or debris on a slope. Prior detection of landslides can save lives and reduce property losses. To ensure public safety and mitigate potential risks, it is essential to correctly identify the triggers of landslides. A landslide trigger refers to any event or condition that causes the downward movement of rock, earth or debris on a slope. These triggers encompass natural events such as heavy rainfall, earthquakes and rapid snowmelt, as well as human activities including construction, deforestation and mining. These activities disrupt slope stability, often leading to significant damage and loss of life. Accurately identifying these triggers is vital for developing effective early warning systems and implementing preventive measures. Our goal in this study is to classify the landslide triggers that have resulted in landslide events. The classes under landslide trigger include construction, continuous rain, downpour, earthquake, flooding, freeze-thaw, mining, monsoon, no apparent trigger, rain, snowfall snowmelt and tropical cyclone.

There are two main categories of landslide susceptibility mapping methods: qualitative and quantitative methodologies. The qualitative approach relies heavily on expert opinions to estimate the potential for landslides based on intrinsic factors. Subjective ratings of susceptibility assessment and variable weighting are also important, making this approach less ideal due to its inherent subjectivity [1]. On the other hand, the quantitative approach, which is more objective due to its reliance on data, uses statistical methods to establish a relationship between landslides and

their controlling factors for a sampled area. This relationship is then extended to the entire research area to assess susceptibility [2]. The quantitative approaches include both statistical and advanced machine learning methods. To address our problem, a dataset based on India with various kinds of landslide-related features (e.g., landslide category, landslide size, fatality count, latitude, longitude, temperature, humidity, soil information etc.) has been considered. Various machine learning models (Logistic Regression, SVM, Naive Bayes, Decision Tree, KNN, Random Forest etc.) have been explored for predicting the class of landslide triggers. In this work, we have introduced a stacked ensemble learning model, where we consider various MLP (Multilayer Perceptrons) variants as base learners and Random Forest as the meta-learner. We assess the models' performance using a range of metrics including Accuracy, Precision, Recall, F1-score and AUC-ROC. This comprehensive evaluation allows us to gauge the effectiveness and reliability of each model in predicting landslide triggers.

1.2 Motivation

Various impacts of landslides have increased in past decades due to the rapid growth of urbanization in the developing world [3]. According to a World Health Organization (WHO) study, landslides affected 4.8 million people and resulted in over 18,000 deaths between 1998 and 2017 [4]. Additionally, the World Bank estimates that over 300 million people live in a 3.7 million square kilometer stretch of inland terrain that is susceptible to landslides [5]. The floods and landslides in China from May to August 2010 caused significant economic damage, ranking second highest among natural disasters with an estimated cost of $18 billion [6]. The significance of these factors underscores the critical need for advanced techniques in landslide analysis. In recent decades, numerous research institutes and governments have made efforts to produce landslide susceptibility maps for disaster prevention [7, 8]. While extensive research exists on landslide prediction and susceptibility mapping, there is a notable gap in studies focused on landslide triggers. These triggers, which include natural events such as heavy rainfall, earthquakes and rapid snowmelt, as well as human activities like construction, mining are crucial for understanding and

predicting landslide events. Accurate detection of these triggers can significantly enhance hazard estimation and risk mitigation. Proper detection of landslide triggers can provide a more accurate estimation of hazards and improve disaster preparedness. This research field requires increased usage of advanced machine-learning algorithms, such as stacked models, to improve the accuracy and reliability of landslide trigger identification and prediction. By focusing on landslide triggers and employing sophisticated machine learning techniques, we can develop more effective models for predicting and mitigating the impacts of landslides.

1.3 Objective

India has several landslide-prone areas, and very few significant study of predicting landslide trigger has been done yet. Identifying landslide triggering factors can also help in taking preventive measures [9]. In addition to fatalities, India also faces a large number of property losses. Large-scale property loss and casualties were caused by landslides in the Himalayan areas [10]. In this context, this research aims to build an efficient model that can predict landslide trigger (construction, continuous rain, downpour, earthquake, flooding, freeze thaw, mining, monsoon, no apparent trigger, rain, snowfall snowmelt and tropical cyclone). This model could help in proper management in those landslide-affected areas.

1.4 Contribution

The Indian Himalayan region, covering more than 12% of the country's land area, is highly prone to landslides. This susceptibility is primarily attributed to its fragile lithology, intricate geological composition, steep slopes resulting in high energy relief and the rugged topography [11]. The main focus of this study is to forecast the trigger class of landslide occurrences. In order to make an accurate prediction a highly detailed dataset based on Indian landslide has been collected by combining landslide locations, date, time, landslide category, landslide trigger, landslide sizes, weather data such as temperature, wind speed, dew, humidity, slope, elevation, NDVI, NDWI , features of soil such as amount of clay and sand etc.

- This study aimed to accurately predict the triggers of landslides in landslide-prone regions of India. It aimed to forecast a range of triggers, including construction activities, prolonged rainfall, heavy downpours, earthquakes, flooding, freeze-thaw cycles, mining activities, monsoon seasons, instances with no discernible trigger, rain events, snowfall snowmelt and tropical cyclones.
- Utilizing stacked-based machine learning algorithms, an advanced predictive model was developed to precisely identify the triggers of landslides.
- The effectiveness of the proposed model was comprehensively assessed by comparing its results with those of numerous baseline models.

1.5 Organization

The remainder of the paper is structured as follows. Chapter 2 provides a thorough analysis of related studies in this field. The problem description for this work is described in Chapter 3. Chapter 3 also contains a description of the study area and the approach of this study, which includes data collection, data preprocessing, and proposed model. Chapter 4 discusses the evaluation of proposed models and comparison with baseline models. In Chapter 5, this research work is concluded, and a future scope is also given.

Chapter 2

Related works

A detailed study of traditional and machine learning based methods for landslide prediction is provided in this section. Since landslides are a frequent natural disaster, a large number of studies are being conducted worldwide by researchers. Different approach and different models have been used in this context.

2.1 Shallow machine learning models for landslide predictions

Chang et al.[12] focuses on landslide susceptibility prediction (LSP) by introducing variant Slope-machine learning models that leverage slope units extracted using the multi-scale segmentation (MSS) method in their research paper. These models consider the heterogeneity of conditioning factors within slope units, leading to more comprehensive LSP modeling. By expanding the original conditioning factors to 38 and utilizing random forest (RF) and multi-layer perceptron (MLP) models, the study demonstrates the superior performance of variant Slope-machine learning models compared to traditional Grid-machine learning models. In their paper, Liu et al. [13] introduce three machine learning algorithms: Random Forest, Gradient Boosted Regression Tree and MultiLayer Perceptron neural network. These algorithms are applied to model shallow landslides near Kvam in Norway. However, the paper does not provide detailed information about the size and representativeness of the dataset used for training and testing the ML models. This information gap could potentially impact the generalizability of the results. Rainfall is a very important parameter of landslide. Often high rainfall induces landslides. How rainfall can

help in landslide prediction in Hong Kong is described by Ng et al. [14]. Several machine learning models such as Logistic Regression, Random Forest, AdaBoost tree, Support Vector Machine and Multi-layer Perceptron are evaluated in this study. Out of all models Random Forest yielded the best result. Although this study focused on the relationship between landslides and rainfall, the most of landslides on naturally occurring topography are not included in the dataset. Understanding the landslide producing factors may be easier with a more thorough inventory. Support Vector Machine (SVM) is very popular machine learning model which is used in landslide susceptibility mapping and it is widely used because it can efficiently handle small datasets.

2.2 Deep learning models of landslide predictions

All the above-mentioned models are shallow machine learning. But now a days several deep learning based models are becoming popular in landslide prediction. A deep convolutional neural network (DCNN-11) is proposed by Wang et al. [15]. This model is trained and tested against three datasets. Though this model produced satisfactory result, the value of the features used in this study were collected after many years of landslide events. So, the prediction could be impacted by this contradiction. Liu et al. [16] proposed a new high-efficiency and high-precision deep learning algorithm called MNTL to identify landslide triggers in the Tibetan Plateau. This method, based on MobileNet-V2 and transfer learning, requires only 30 samples per class to achieve 94% forecast accuracy. Compared to six state-of-the-art deep learning classification methods (VGG-16, VGG-19, ResNet-50, ResNet-101, Inception and MobileNet-V2), the proposed method demonstrated superior convergence speed and generalization capability. The lightweight network designed in this study can be embedded into portable mobile devices, reducing research costs and facilitating the rapid dissemination and application of findings. However, despite its high accuracy, the model has limitations; it was only applied to classify landslides triggered by rainfall and thawing effects. It remains unclear if the method is applicable to other types of landslides. Aguilera et al. [17] proposed a Hierarchical Neural Network (HNN) for landslide prediction Kathmandu, Nepal. This hybrid

model can predict both the landslide sizes and landslide locations. So, this model is more advanced than other LSM models but it is blind to temporal frequency of each landslide sizes. S.Sreelakshmi et al. [18] suggested a Deep Convolutional Neural Network for classifying landslide data. As baseline models, six shallow machine learning models are used: Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN) and Multi-Layer Perceptrons (MLP), a deep learning model. The authors claimed that advanced techniques like DCNN can be an useful method in landslide prediction using a number of performance matrices.

2.3 Hybrid machine learning models for landslide predictions

Hybrid machine learning models are also popular at this field in addition to shallow and deep learning models. Trong-An Bu et al. [19] presents a method for detecting landslide location in satellite images using a combination of deep learning and image transform algorithms. The proposed algorithm uses a convolution neural network for image classification and a transformation algorithm called Hue-Bi-dimensional empirical mode decomposition (H-BEMD) to accurately identify landslides under different lighting conditions. The study reports an accuracy of up to 96% in the classification process and almost absolute accuracy in landslide location. Additionally, the paper presents a method for predicting landslide size scaling and direction. However, the results are affected by cloud cover with a lot of coverage, the process of identifying objects is noisy. Artificial Neural Network (ANN) is very common in landslide prediction. Particle swarm optimisation (PSO) was used with an artificial neural network (ANN) by Moayedi et al. [20] to enhance the performance of the ANN in landslide susceptibility mapping. The authors claim that when compared to ANN, the PSO-ANN technique demonstrated greater reliability. Dou et al. [21] evaluate the predictive capability of hybrid ensemble SVM algorithms, including bagging, boosting and stacking, for modeling catastrophic rainfall-induced landslides in Northern Kyushu Island, Japan, at the watershed scale. The landslide

susceptibility maps (LSMs) were validated using the area under the receiver operating characteristic curve (AUC). Validation results showed AUC values ranging from 0.74 to 0.91 for the four models (SVM-Stacking, SVM, SVM-Bagging and SVM-Boosting), with the SVM-boosting model outperforming the others. However, the study's focus on the Northern parts of Kyushu Island limits the generalizability of the results to other regions. Additionally, the study considers only rainfall-induced landslide susceptibility and does not account for other triggering factors. A rough set theory linked with machine learning models was proposed by Kainthura et al. [22]. Five alternative models are trained and tested on the landslide dataset of Uttarakhand, India: Bagging (HBRS), the Bayesian Network (HBNRS), XGBoost (HXGBRS), Random Forest (HRFRS) and Backpropagation Neural Network (HBPNNRS). According to the authors of this study, the HXGBRS model provides the most accurate result out of all models. Tien Bui et al. [23] employed the logistic model tree (LMT), logistic regression (LR), stochastic gradient descent (SGD) and functional tree (FT) to forecast landslides in the Sarkhoon watershed, Iran. A hybrid model of Stochastic Gradient Descent and AdaBoost (ABSGD), which is employed in addition to the models mentioned above to improve the accuracy of landslide prediction models, performed better than other baseline models. In order to forecast landslides in Thailand's Chiang Rai Province, Tengtrairat et al. [24] used five machine learning models, including logistic regression (LR), artificial neural network (ANN), gated recurrent units (GRU), long short term memory (LSTM) and bidirectional LSTM (Bi-LSTM). Two hybrid models, Bi-LSTM with logistic regression (Bi-LSTM-LR) and Bi-LSTM with random forest (Bi-LSTM-RF), are also utilised in addition to machine learning models. The authors claim that the Bi-LSTM-RF model produced the best outcome. Using four machine learning models—Random Forest (RF), K-Nearest Neighbor (KNN), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM) Adnan et al. [25] emphasised the discrepancy of landslide predictions. This discrepancy indicates the unreliability of various machine learning models' forecasts of landslides. The authors stated that they were able to reduce the uncertainties and enhance both prediction accuracy and spatial disagreement by merging the findings from these four models. Unfortunately, this study uses low resolution DEM data. The mapping of landslide susceptibility may benefit from high

TABLE 2.1: Existing literature on landslide prediction

Author	Year	Model	Description/ Contribution	Data Source / Study Area	Performance metrics
S.Sreelakshmi et al. [18]	2023	SVM, LR, NB, RF, DT, KNN, MLP, DCNN	Using a variety of machine learning and deep learning techniques, classify landslides (both landslides and non landslides). The proposed model is DCNN.	Publicly available labelled dataset	Confusion matrix, Accuracy
Chang et al.[26]	2022	RF, MLP	Introduces slope unit-based machine learning models for landslide susceptibility prediction, utilizing the multi-scale segmentation method for slope unit extraction.	Chongyi County in China	AUC, FR
Kainthura et al. [22]	2022	HRFRS, HXGBRS, HBRS, HBNRS, and HBPNNRS	In this research, five hybrid machine learning (ML) models are employed to enhance the prediction capabilities of ML models. All models produced encouraging results, although model HXGBRS performed better than other models.	Uttarkashi, Uttarakhand (India)	Accuracy, Sensitivity, Precision, Specificity, F1-Score, AUC
Azarafza et al. [27]	2021	SVM, CNN-DNN, MLP, LR, BNB, GNB and DT	The deep learning-based landslip susceptibility mapping was the main topic of this work. Here, the CNN-DNN model is suggested.	Isfahan province, central Iran	Accuracy, Precision, Recall, F1-score, RMSE, AUC, MAPE, MSE
Ng et al.[28]	2021	LR, RF, adaboost, SVM, and MLP)	The paper focuses on building a landslide prediction model in Hong Kong using real-time rainfall data and various controlling factors.	Hong Kong	Accuracy, recall, precision, AUC and OA
Rane et al.[29]	2021	MLE, ISO clustering, and RF	The ISO algorithm surpasses both the ML and RF algorithms in terms of Area Under the Curve (AUC).	Uttarakhand in north	accuracy, sensitivity, specificity
Tengtrairat et al. [24]	2021	LR, GRU, ANN, LSTM, and Bi-LSTM	For landslide prediction, five models and the combined BiLSTM-LR and BiLSTM-RF were utilised. Other models were outperformed by BiLSTM-RF.	Chiang Rai, Thailand	Accuracy, PPR, AUC, NPR
Wang et al. [15]	2021	Boosting, LR, RF, SVM, CNN, DCNN	In this research, the DCNN-11 model, which stands for Deep Convolutional Neural Network with 11 layers, is suggested for landslide prediction.	Lantau Island	Accuracy, Specificity, Precision, FPR, Recall, F1-Score, MCC
Liu et al.[13]	2021	RF, GBRT, MLP	Objective of study is to demonstrate that ensemble machine learning techniques can be applied to a large, heterogeneous area containing a number of diverse landslides GBRT yielded the most promising results for the spatial prediction.	Kvam in Norway	Accuracy, AUC, POD, POFD

resolution data.

Most of the studies on landslide prediction are on identifying landslide prone areas, risk identifications. Relation between landslide causing factors with landslides are also done in some study. But research on classifying the trigger of landslides is quite uncommon in this field. So, in this work, we are interested in forecasting all the trigger of landslides in India's landslide-prone locations, which are primarily divided into twelve classes: *construction activities*, *prolonged rainfall*, *heavy downpours*, *earthquakes*, *flooding*, *freeze-thaw cycles*, *mining activities*, *monsoon seasons*, *instances with no discernible trigger*, *rain events*, *snowfall snowmelt* and *tropical cyclones*.

TABLE 2.1(continued) : Existing literature on landslide prediction

Author	Year	Model	Description/ Contribution	Data Source/ Study Area	Performance metrices
He et al. [30]	2021	SVM, RF, MLP, DLNN	This study examined how including discontinuities in machine learning models can improve their ability to forecast landslides.	North Coast of Cornwall, UK	ROC-AUC, Confusion matrix
Zhang et al. [31]	2020	ELM, WCA-BPNN, BPNN, SVM,	In comparison to the conventional BPNN model, the WCA-BPNN model is presented to achieve faster convergence and improved prediction.	Three Gorges Reservoir, China	Accuracy, RMSE, MAPE
Achour et al. [32]	2020	SVM, RF, BRT	The use of machine learning in landslide prediction is discussed in this work. Here, Random Forest fared better than competing models.	NE Algeria	AUC
Adnan et al. [25]	2020	KNN, RF, SVM, and MLP	It is merged the four models (KNN, MLP, RF, and SVM) to improve spatial agreement and prediction accuracy.	Cox's Bazar district, Bangladesh	Accuracy, Recall, Precision, AUC, F1-Score
Yi,Zhang[33]	2020	LandsNet	Introduces a new deep-learning-based method for accurately and reliably identifying and mapping earthquake-triggered landslides using single-temporal optical RapidEye images.	China	Precision, Recall, F1 score (F1), overall accuracy (OA) and Kappa coefficient (KAPPA)
Nhu et al. [34]	2020	ADTree, AB, AB-ADTree	For landslide prediction, we employ the AB, ADTree, and AB-ADTree models. AB fared better than other models.	Cameron Highlands, Malaysia	Sensitivity, RMSE, Spesificity, Accuracy
Kuradusenge et al. [35]	2020	LR, RF	This study demonstrates how antecedent rainfall can improve landslide model prediction.	Ngororero District, Rwanda	Recall, Accuracy, FPR, ROC-AUC, FNR
Dao et al. [36]	2020	QDA, MLP, FLDA, DL	In this research, a deep learning model for landslide susceptibility is proposed.	Muong Lay distric, Vietnam	Accuracy, AUC, RMSE
Fanos et al. [37]	2019	Stacking ensemble	The study focuses on utilizing LiDAR-derived landslide conditioning to assess the performance of laser scanning data for landslide probability, offering an alternative to traditional structural and geotechnical factors.	Kinta Valley, Malaysia	ROC, PRC
Tien Bui et al. [23]	2019	GD, LMT, FT, LR, ABSGD	To forecast the spatial distribution of landslides, an ABSGD hybrid functional machine learning model is presented.	Sarkhoon watershed, Iran	Accuracy, RMSE, Sensitivity, Specificity, AUC
Pham et al. [38]	2019	ADTree, RF, Bagging	A hybrid machine learning model called RS-ADT was introduced in this paper.	Pithoragarh district, Uttarakhand,India	TPR, FPR, Recall, ROC-AUC, Precision, F-measure
Moayedi et al. [20]	2018	PSO-ANN	To enhance the performance of ANN, the author of this research applied an optimisation approach like PSO.	Layleh valley, Kermanshah,Iran	RMSE, Accuracy
Kadavi et al. [39]	2018	AdaBoost, Bagging, LogitBoost	This study aims to create a map of landslide susceptibility using an ensemble of machine learning models.	Gangneung City, South Korea	AUC

Chapter 3

Methodology

In this chapter, problem statement and studyarea are described in Section 3.1 and Section 3.2 respectively and then the proposed approach is described in three sections namely *data collection and description*, *data preprossing* and *proposed model*.

Figure 3.2 describes all the steps which are followed in this study. Using landslide locations raw data of landslide causing factors are collected from different source. A final dataset is obtained by combining all the collected data. Section 3.3 describes the data collection and description in detail. But it is necessary to preprocess the dataset before using it in model building. As data are collected from various sources, there was missing value problem in the combined dataset. Proper steps were taken to make a complete dataset. Categorical variables are also transformed into numerical variable to make the dataset more understandable by the machine learning models. This data preprocessing part is described in Section 3.4 in detail. The final dataset is divided into two parts: a training set and a test set, using 5-fold cross-validation. The training set is used to train the models, while the test set is used for the final evaluation of the models. Description of the proposed model and evaluations by using various performances matrices are described in Section 3.5 and Chapter 4 respectively.

3.1 Problem Statement

Predicting landslide triggers is a complex endeavor, requiring the analysis of various contributing factors. Multiple elements are involved in this prediction process. To efficiently forecast landslide triggers, we employ a set of features denoted by X, including parameters such as Landslide category, Landslide size, Fatality count,

Latitude, Longitude, Temperature, Clay content etc. Our aim is to devise a predictive model represented by $f(.)$ such that $y = f(X)$ where y is the predicted landslide trigger class which includes construction, continuous rain, downpour, earthquake, flooding, freeze thaw, mining, monsoon, no apparent trigger, rain, snowfall snowmelt, tropical cyclone. The function f belongs to a set of candidate models, including Logistic Regression, Decision Tree, Naive Bayes, SVM, Random Forest, XG Boost, ANN and Stacking ensemble with different base and meta learner.

3.2 Study Area

Three major landslide prone areas of India (*north India, northeast India* and *southwest India*) are considered as the study area for this research. The locations of these areas are given in this section.

North India, including Jammu and Kashmir, Himachal Pradesh, Punjab, Haryana and Uttar Pradesh has been chosen as the study area for this research. This area covers almost 178196.58 km^2 area, which extends from the latitude of 28.25N to 34.724N and longitude of 74.5E to 80.862E. Similarly, another two landslide-prone regions are included into the study area to build the dataset. The North East India covers Sikkim, Assam, Meghalaya, Manipur, Mizoram, Nagaland and Arunachal Pradesh. This area covers around 234976.24 km^2 area, extending from the latitude of 22.27N to 28.07N and longitude of 88.22E to 96.87E. Maharashtra, Karnataka, Tamil Nadu and Kerala are major landslide prone areas of South West India. This part covers around 334215.45 km^2 area and extends from the latitude of 8.27N to 20.36N and longitude of 72.68E to 77.99E. These three sections cover all the regions that frequently undergo deadly landslides that cause significant damage and casualties. Figure 3.1 represents all the landslide locations that are used in this study.

3.3 Data collection and description

The most important part of this study is to obtain the landslide locations. These location are obtained by Global Landslide Catalogue (GLC) [40]. The related CSV file is open to access and can be downloaded from the website of US Government's open

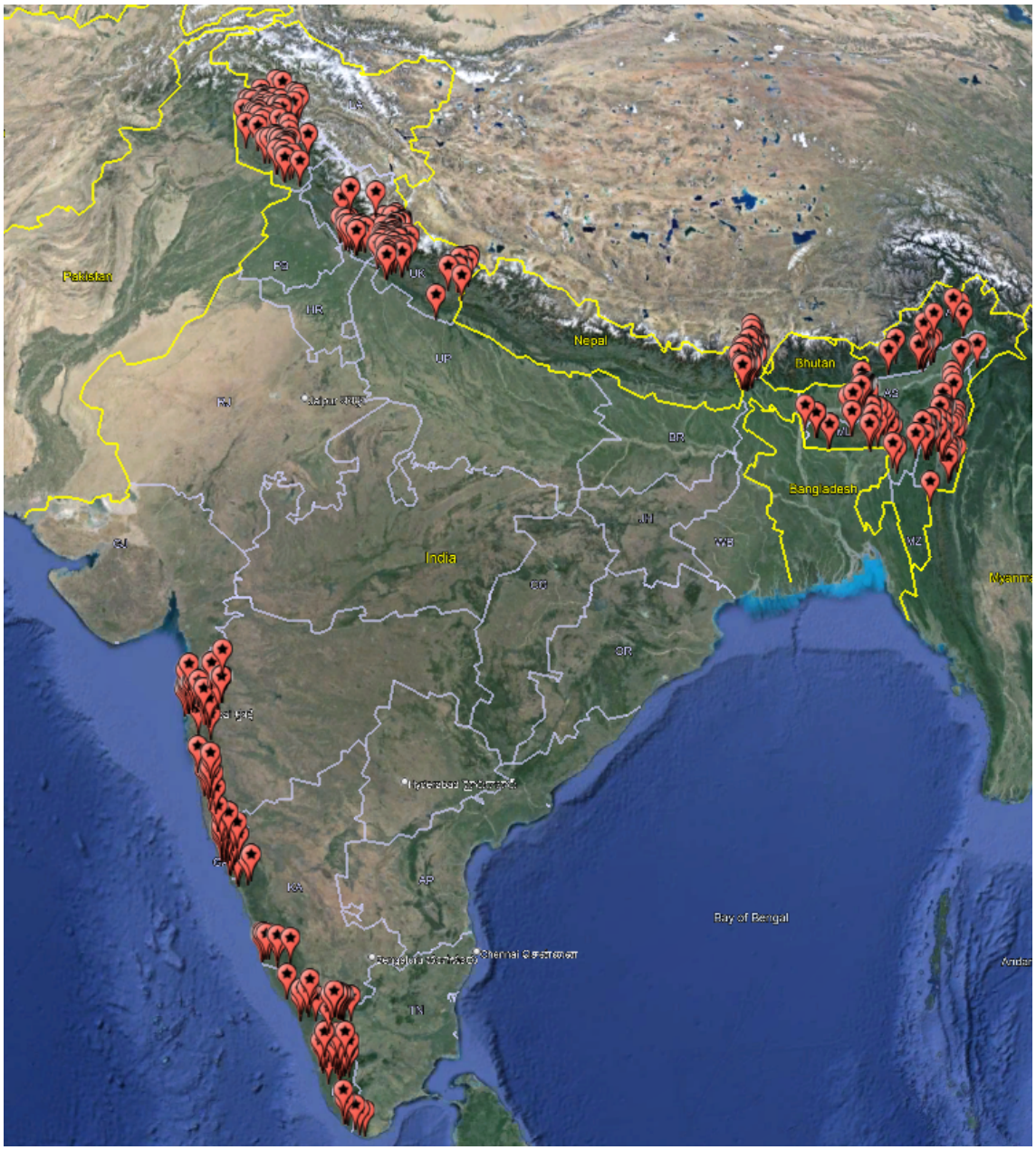


FIGURE 3.1: Locations in India where landslides have been occurred

data (https://catalog.data.gov/dataset/global-landslide-catalog-export). This CSV file contains all landslide relation data such as, date of landslides, location of landslide or latitude, longitude, landslide trigger (such as downpour, earthquake, mining etc.), landslide sizes etc. VisualCrossing Weather API (https://www.visualcrossing.com/weather-api) are used to obtain different weather data such as temperature, wind speed, dew, humidity, precipitation etc. To get location specific data, date, time and latitude, longitude of landslide events are used. Some geomorphic factors such as elevation, slope, aspect, vegetation index (NDVI), water index (NDWI) etc. are obtained from geospatial analysis platform Google Earth Engine [41]. Features of soil such as chemical composition of soil, organic contents, amount of clay, sand etc. of an area are important factors for analysing landslide sizes. To get such data SoilGrids API (https://rest.isric.org/soilgrids/v2.0/docs) are used. All these data are then combined to get a single dataset which is further preprocessed to obtain the final dataset for model building. The next section describes data preprocessing part in detail.

3.4 Data preprocessing

Data preprocessing and feature selection are very important part of dataset making. All important landslide causing factors which are found after studying various literature, are included in the dataset. All this factors are described in Table 3.1. Data preprocessing techniques are employed to transform raw data into a suitable format for predictive models, thereby enhancing their accuracy and efficiency. In this study, the raw landslide dataset contains missing attributes, which are addressed through cleaning to remove these missing attributes. Subsequently, label encoding is utilized to convert categorical data into numerical form. After this step, the dataset is divided into features (X) and the target variable (y). Notably, the features encompass various attributes such as landslide category, size, fatality count, latitude, longitude, temperature, clay content etc. The target variable comprises 12 classes representing landslide triggers, including construction, continuous rain, downpour, earthquake, flooding, freeze thaw, mining, monsoon, no apparent trigger, rain, snowfall

TABLE 3.1: Features and descriptions

Features	Description
maxslope	Highest slope at the landslide's origin
avgslope	Average slope of various angles relative to the origin
slope	Slope at landslide location
LC	Average value of Land Cover, using landslide occurrence time
LST	Day time average of Land Surface Temperature, using landslide occurrence time
gHM	The average value of human modification of terrestrial lands worldwide, using landslide occurrence time
precipitation	Total precipitation
NDVI	Normalized Difference Vegetation Index's average value, using landslide occurrence time
NDWI	Normalized Difference Water Index's average value, using landslide occurrence time
SPI	Stream Power Index
aspect	The slope's elevation direction
elevation	Elevation
landslide_category	Type of landslide (landslide, mudslide, rock fall etc.)
landslide_trigger	Landslide trigger (downpour, flooding, rain etc.)
precipcover-2, precipcover-1, precipcover-0	Precipitation coverage prior two days, one day, and the day of the landslide
temp-2, temp-1, temp-0	Temperature prior two days, one day, and the day of the landslide
maxt-2, maxt-1, maxt-0	Highest temperature prior two days, one day and the day of the landslide
wspd-2, wspd-1, wspd-0	Wind speed prior two days, one day and the day of the landslide
mint-2, mint-1, mint-0	Lowest temperature prior two days, one day, and the day of the landslide
precip-2, precip-1, precip-0	Precipitation prior two days, one day and the day of the landslide
sealevelpressure-2, sealevelpressure-1, sealevelpressure-0	Sea level pressure prior two days, one day, and the day of the landslide
dew-2, dew-1, dew-0	Dew prior two days, one day and the day of the landslide
humidity-2, humidity-1, humidity-0	Humidity prior two days, one day, and the day of the landslide
O	Elevation where the landslide occurred.
N, NE, NW, S, SE, SW, E, W	Elevations at the points surrounding the original location of landslide
clay0-5, clay5-15, clay15-30, clay30-60, clay60-100, clay100-200	Clay content particles <0.002 mm in the fine earth fraction at depths 0-5 cm, 5-15cm, . . . , 100-200 cm
bdod0-5, bdod5-15, bdod15-30, bdod30-60, bdod60-100, bdod100-200	Bulk density of fine earth fraction at depths 0-5 cm, 5-15cm, . . . , 100-200 cm
cec0-5, cec5-15, cec15-30, cec30-60, cec60-100, cec100-200	Cation Exchange Capacity of the soil at depths 0-5 cm, 5-15cm, . . . , 100-200 cm
cfvo0-5, cfvo5-15, cfvo15-30, cfvo30-60, cfvo60-100, cfvo100-200	Volumetric Fraction of Course Fragments > 2mm at depths 0-5 cm, 5-15cm, . . . , 100-200 cm
n0-5, n5-15, n30-60, n60-100, n100-200	Levels of nitrogen at depths 0-5 cm, 5-15cm, . . . , 100-200 cm
ocd0-5, ocd5-15, ocd15-30, ocd30-60, ocd60-100, ocd100-200	Organic Density of organic carbon at depths of 0-5 cm, 5-15cm, . . . , 100-200 cm
ocs0-30	Organic Carbon Stocks at depth 0-30 cm
phh2o0-5, phh2o5-15, phh2o15-30, phh2o30-60, phh2o60-100, phh2o100-200	pH of the soil at depths of 0-5 cm, 5-15cm, . . . , 100-200 cm
sand0-5, sand5-15, sand15-30, sand30-60, sand60-100, sand100-200	Sand particle percentage > 0.05 mm in the fine earth fraction at depths of 0–5 cm, 5–15 cm, . . . , 100–200 cm
silt0-5, silt5-15, silt15-30, silt30-60, silt60-100, silt100-200	Silt particles that are > 0.002 mm and < 0.05 mm in size in the fine earth fraction at depths of 0–5, 5–15, . . . , 100–200 cm
soc0-5, soc5-15, soc15-30, soc30-60, soc60-100, soc100-200	Fine earth percentage soil organic carbon concentration at depths of 0–5 cm, 5–15 cm, . . . 100–200 cm

snowmelt and tropical cyclone. The integrated dataset from multiple sources contains missing values, which are addressed by removing them from the model to enhance performance. Additionally, to tackle class imbalance, SMOTE [42] is applied. SMOTE randomly selects instances from the minority class and generates synthetic

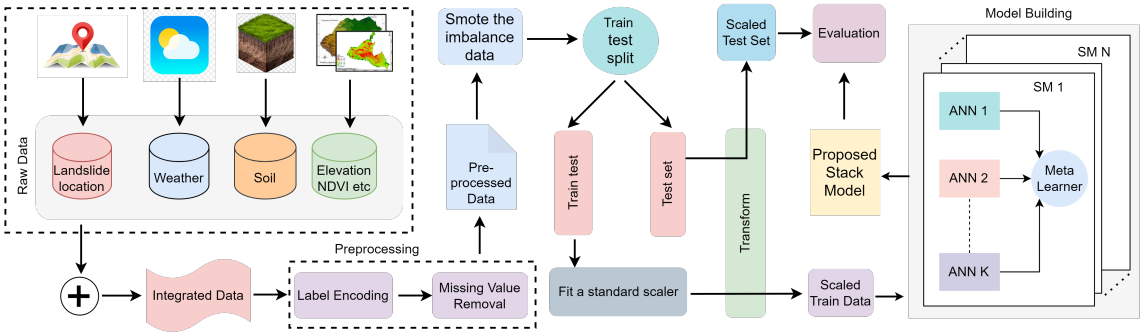


FIGURE 3.2: Work flow diagram of the study

examples in feature space using k-nearest neighbors. This technique is crucial due to significant class imbalance within the dataset. Furthermore, the dataset is split into training and testing sets, with approximately 75% of the data allocated for training and the remaining portion for testing. The total dataset comprises 3991 samples, with 2993 samples used for training and 998 for testing. Finally, feature scaling is conducted to normalize features with differing ranges, ensuring fair comparison and effective analysis by machine learning models.

3.5 Proposed model

This study's complete methodology is summarised in Figure 3.2. We begin by building the dataset, which involves collecting data on landslide locations (latitude, longitude, sea level etc.), weather information (temperature, humidity, precipitation, dew etc.), soil characteristics (clay content, fertility etc.), elevation and NDVI. After gathering all the raw data, it is integrated into a final dataset. The integrated data then undergoes preprocessing, which includes label encoding to convert categorical variables into numerical format and the removal of missing values. Since the dataset is imbalanced, we utilize the imbalanced-learn library, available at (https://pypi.org/project/imbalanced-learn/), to implement the SMOTE algorithm. The parameter *k_neighbors* is set to 3, with all other hyperparameters left at their default values. Following this preprocessing, Dataset are divided into train set and test set with 5 fold cross validation. To ensure the robustness of the model, the training data is further preprocessed by standardizing it using StandardScaler, which calculates the mean and standard deviation of each feature and scales the data accordingly.

This step ensures that all features are on the same scale, preventing any single feature from dominating the modeling process due to larger magnitudes. The datasets are used to train and evaluate a variety of machine learning models, including stacked models.

After transformation, the training set is ready for model training. A stacking ensemble model with k (in our case $k = 16$) MLP layers and various meta-learners is used for classification. The final step involves evaluating the model's performance using various metrics such as accuracy, precision, recall, f1-score, the AUC and the confusion matrix. A total of sixteen models including baseline models and stacked model are trained for landslide trigger prediction. The stacked model appear to produce the best results overall.

3.5.1 Stacked machine learning model

Stacking in machine learning is a technique to ensemble different classification and regression models. Ensembling [43] is the way to use more than one model for prediction purpose. The reason behind using the ensembling technique is, often it produces better result than the base learners taken separately. There are three types [44] of ensembling technique i.e. bagging [45], boosting [46] and stacking [47]. The main difference of stacking with bagging and boosting is, stacking uses heterogeneous weak learners while other two use homogeneous weak learners.

We have introduced a Stacked ensemble learning technique shown in Figure 3.3 for landslide trigger prediction as the proposed model. Stacked ensemble learning technique that combines multiple base models to create a meta-model. This meta-model makes predictions based on the collective decisions of its constituent models. The proposed model is designed and implemented to enhance predictive performance by leveraging the strengths of multiple neural network base models. The process begins with loading the original dataset. After data preprocessing, which includes oversampling with SMOTE to address class imbalance, neural network models with distinct architectures are trained and saved. The predictions from these base models are then used as input features for the meta-learner in the ensemble. Through 5 fold cross-validation, the stacked ensemble is trained on the concatenated predictions of the base models and evaluated on a test set.

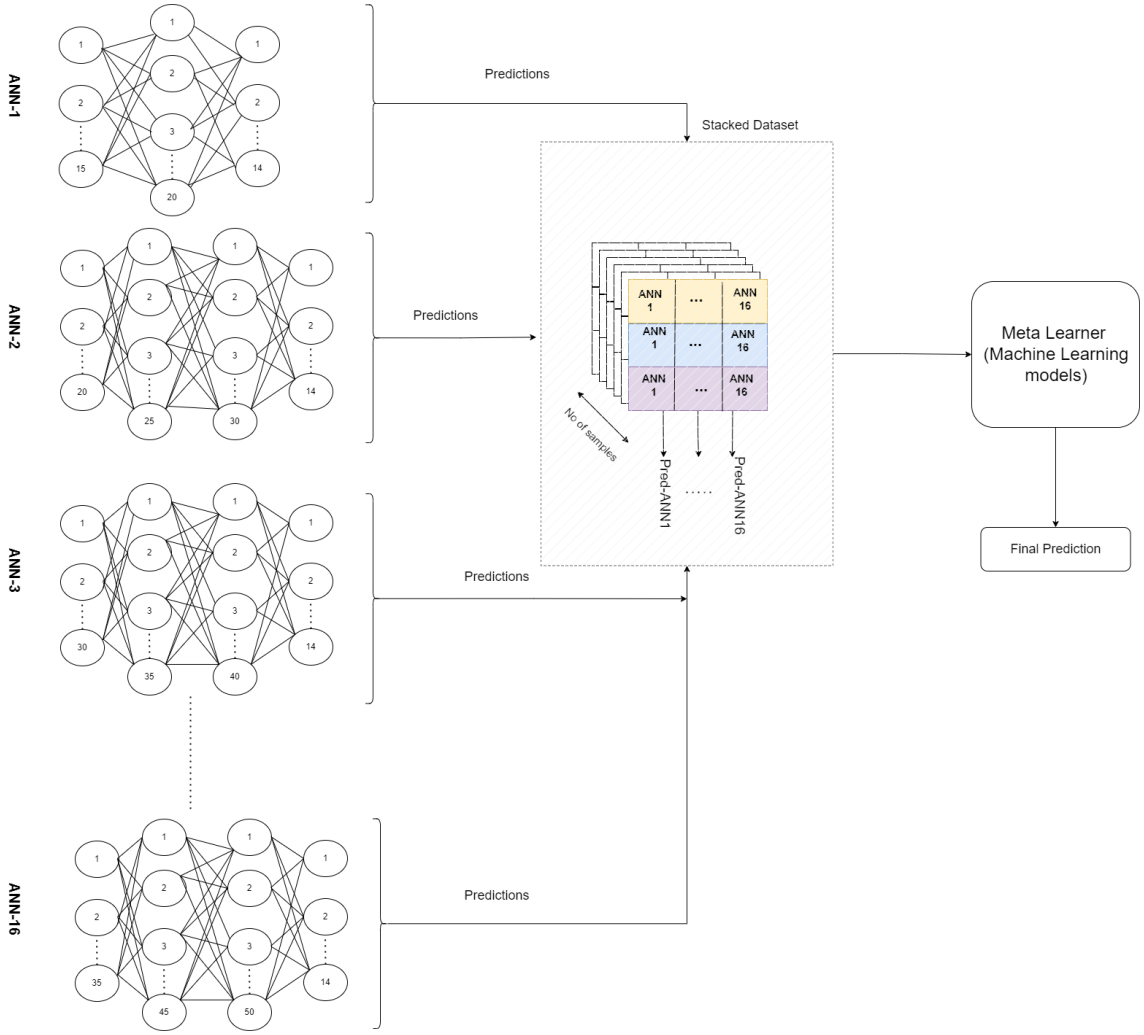


FIGURE 3.3: Architecture of proposed stacked ensemble learning model

TABLE 3.2: Specifications of Artificial Neural Network Models

No of ANN	Layers	Neurons per Layer	Activation Function	Epochs
1	3	15, 20, 14	ReLU, ReLU, Softmax	100
2	4	10, 15, 25, 14	ReLU, ReLU, ReLU, Softmax	150
3	4	20, 25, 30, 14	ReLU, ReLU, ReLU, Softmax	180
4	4	30, 35, 40, 14	ReLU, ReLU, ReLU, Softmax	200
5	4	40, 50, 55, 14	ReLU, ReLU, ReLU, Softmax	250
6	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
7	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
8	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
9	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
10	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
11	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
12	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
13	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
14	4	35, 45, 50, 14	ReLU, ReLU, ReLU, Softmax	300
15	4	35, 45, 50, 14	ReLU, Sigmoid, ReLU, Softmax	300
16	4	35, 45, 50, 14	ReLU, Sigmoid, ReLU, Softmax	300

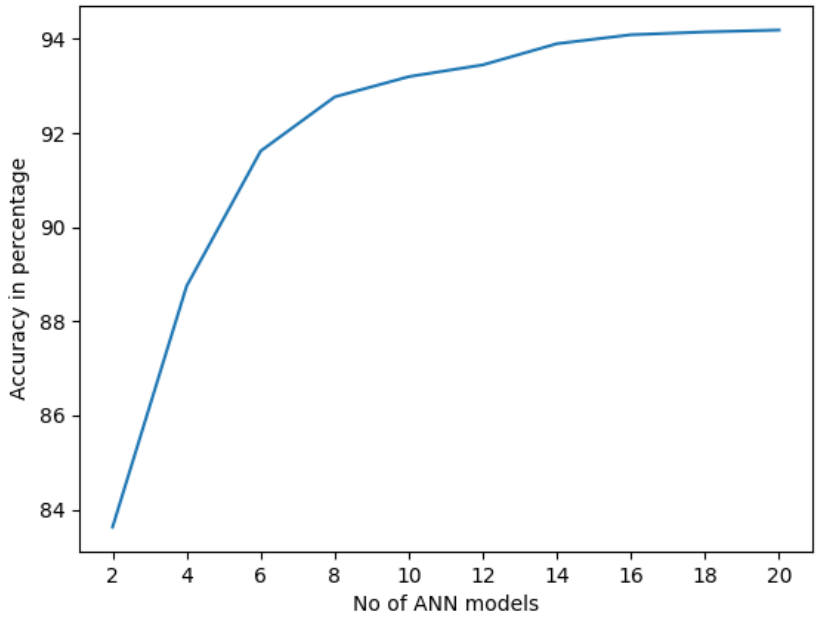


FIGURE 3.4: Number of base model vs accuracy plot

In our base model, there are sixteen artificial neural network models for the classification task. The input data is split into training and testing sets for five different random states. Each ANN follows a sequential architecture, consisting of dense layers with varying numbers of neurons and activation functions. The first fourteen models employ rectified linear unit (ReLU) activation in their hidden layers, while the fifteenth and sixteenth models incorporate sigmoid activation. These models differ in terms of layer architecture, with the number of hidden layers being either two or three and the number of neurons increasing progressively. The models are trained using the Adam optimizer and sparse categorical crossentropy as the loss function, with varying epochs for each layer. After training, the models are saved to separate files under the "north" directory. The diversity in architecture and random state variations aims to capture a wide range of patterns in the training data, potentially enhancing the ensemble's robustness and predictive performance. The specification of all the ANN layers are given in Table 3.2. For meta learner we have used Random forest, SVC RBF, SVC linear, gaussian naive bayes, logistic regression, KNN, Gradient boosting, lightGBM, decision tree, XG Boost.

The Figure 3.4 validate the reason of having 16 layer ANN as the base model. While trying with various number of ANN we can clearly noticed that the accuracy

curve is starting converging in the 16^{th} layer of ANN with highest accuracy.

Chapter 4

Result and Analysis

This section analyzes the performance of the proposed model with a number of baseline models. For this comparison, many evaluation measures including accuracy, precision, recall, F1 score and AUC are considered. To prepare the models the dataset is divided into training set and test set. Training set is used to train our models and test set is used for final evaluation of our models. The original dataset is imbalanced, so it was balanced using the oversampling technique SMOTE [48]. The models were then trained on this balanced dataset, and the results obtained from this process are discussed in this section.

4.1 Experimental setup

The experimental setup involves conducting machine learning or deep learning experiments on a system equipped with a 12th Gen Intel(R) Core(TM) i5-12500H processor running at 2.50 GHz and 8.00 GB of RAM (7.63 GB usable). The operating system is a 64-bit architecture. For software and environment setup, Python 3.11 and relevant libraries like NumPy, pandas, scikit-learn, TensorFlow and Keras are installed.

4.2 Baselines and Performance metrics

To assess the efficacy of the proposed model, it is compared with different baseline models including Random Forest(RF) [49], Decision Tree (DT) [50], Support Vector Machines(SVM) [51], K-nearest Neighbours(KNN) [52], Gradient Boosting(GB) [53],

Logistic Regression(LR) [54], Naive Bayes (NB) [55] and XG Boost(XGB) [56]. Here's a brief description of each model-

- Decision Tree (DT): Decision trees are hierarchical models that recursively split data based on feature values, leading to a final decision at the leaves. They're intuitive and easy to interpret, making them useful for understanding decision-making processes.

- Random Forest (RF): Random Forest is an ensemble learning method that constructs multiple decision trees during training and aggregates their predictions. By averaging or voting on these predictions, Random Forest improves robustness and reduces overfitting compared to individual decision trees.

- Support Vector Classification (SVC): SVC is a supervised learning algorithm that separates data points into different categories by finding the optimal hyperplane that maximizes the margin between classes. It's effective in high-dimensional spaces and is particularly useful when dealing with complex datasets.

- K-Nearest Neighbor (KNN): KNN is a non-parametric, instance-based learning algorithm that classifies new data points based on the majority class of their nearest neighbors in the training set. It's simple yet effective, especially in cases where decision boundaries are irregular or difficult to define.

- Logistic Regression (LR): Despite its name, logistic regression is a linear model used for binary classification. It models the probability of a certain class using the logistic function, making it suitable for estimating probabilities and making predictions with confidence levels.

- Naive Bayes (NB): Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, assuming independence among features. Despite its simplicity and strong assumptions, Naive Bayes often performs surprisingly well on a wide range of classification tasks.

TABLE 4.1: Performance Comparison of the Proposed Model with Baseline ML Models

Metrics	Proposed	LR	RF	DT	SVC	NB	KNN
accuracy	0.946	0.7	0.91	0.84	0.88	0.34	0.88
precision macro	0.945	0.645	0.902	0.825	0.87	0.328	0.857
precision micro	0.946	0.7	0.906	0.835	0.88	0.336	0.877
recall macro	0.946	0.66	0.906	0.834	0.88	0.334	0.877
recall micro	0.946	0.702	0.906	0.835	0.88	0.336	0.877
f1-score macro	0.945	0.66	0.903	0.829	0.874	0.297	0.855
f1-score micro	0.946	0.702	0.906	0.835	0.88	0.336	0.877
auc macro	0.946	0.905	0.992	0.909	0.988	0.762	0.968
auc micro	0.946	0.946	0.997	0.91	0.996	0.78	0.97

- Boosting Techniques: Boosting algorithms, such as XGBoost, sequentially train weak learners to correct the errors of their predecessors. By combining multiple weak models into a strong learner, boosting methods achieve high accuracy and robustness, making them popular choices in various machine learning tasks.

- Gradient Boosting (GBRT): Gradient Boosting is an ensemble learning technique that builds a series of weak learners, typically decision trees, and combines them to improve accuracy and reduce overfitting. It optimizes a loss function by iteratively adding models, focusing on minimizing errors, and gradually improving predictions.

Accuracy (Eq.) 4.1, precision (Eq.) 4.2, recall (Eq.) 4.3, F1-score (Eq.) 4.8 and AUC (Eq.) 4.1 are the five performance metrics used to evaluate each of these models. Table 4.1 represents the performance comparison between the proposed model with the baseline models.

The time that has been taken by the baselines are shown in Table 4.2. Notably, the Random Forest model exhibits the longest training time among the models, taking 64,523.78 milliseconds, while the KNN model displays the shortest training time at 32.55 milliseconds but incurs a substantial testing time of 1732.61 milliseconds due to its lazy learning approach. In terms of testing, the SVM model has the lengthiest test time, requiring 9,614.93 milliseconds, whereas the LR model shows the fastest test time at 31.8 milliseconds.

TABLE 4.2: Baseline Models' Train and Test Times (in milliseconds)

ML Model	Train Time	Test Time
LR (Logistic Regression)	3434.26	31.8
Random Forest	64523.78	245.39
Decision Tree	17019.96	32.3
SVM (Support Vector Machine)	46912.84	9614.93
Naïve Bayes	93.99	60.92
KNN (K-Nearest Neighbors)	32.55	1732.61
XGB (XGBoost)	32873.1	221.28

4.3 Result and Discussion

This section represents detailed analysis of the performance of the proposed model and this performance is compared with different baseline models. We have proposed stacked ensemble learning for classification task. The optimal specification settings are presented in Table 3.2 where we have mentioned the number of ANN, hidden layer per ANN, units, activation function and number of epochs each layer. The outcome of various meta learner which are Random forest, SVC rbf, SVC linear, gaussian naive bayes, logistic regression, KNN, Gradient boosting, LightGBM, Decision tree, XG Boost are given in Table 4.3. The evaluation metrices that we have used are Accuracy [57], Precision macro [58], Precision micro, Recall macro [58], Recall micro, F1 macro [59], F1 micro, AUC Macro [12], AUC Micro. The percentage of correct predictions made by the classifier out of all the predictions is known as accuracy. The mathematical formula of accuracy is represented by Eq. 4.1. The classifier's expected number of true positive, true negative, false positive and false negative predictions is denoted by the symbols tp, tn, fp and fn. Similarly, the terms precision, recall and F1-score are represented by Equations 4.2, 4.3 and 4.8, respectively. Precision and recall for multi-class classification cannot be calculated directly using Equations 4.2 and 4.3. This issue is dealt with using macro average and micro average, two averaging techniques. Calculating precision and recall for the macro average requires adding together all tp, tn, fp, and fn values (Eqs. 4.6 and 4.7,where class $c \in \{1, 2, ...N\}$, N is the number of classes). We utilise the simple arithmetic mean of the per-class recall value and precision as the micro average (Eq. 4.4, Eq. 4.5). In a similar way, we may obtain the macro and micro averages of the F1-score (Eq 4.9, Eq. 4.10). Macro averaging of the performance matrices is used for this

study. AUC-ROC curve is an essential measurement for classification problems. The Receiver Operating Characteristics (ROC) curve can be created by plotting the false positive rate (FPR) (Eq. 4.11) on the x-axis and the true positive rate (TPR or recall) (Eq. 4.3) on the y-axis at various threshold settings. Higher AUC score denotes that the model can discriminate between classes more effectively. Confusion matrix is also employed to assess the performance of the baseline models and the proposed model in addition to all these numerical measures. An $N \times N$ matrix called a confusion matrix is used to evaluate the performance of any classification issue, where N is the total number of predicted classes. This matrix compares the actual and expected target values.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{4.1}$$

$$Precision = \frac{tp}{tp + fp} \tag{4.2}$$

$$Recall = \frac{tp}{tp + fn} \tag{4.3}$$

$$Precision_micro = \frac{\sum_c \frac{tp_c}{tp_c + fp_c}}{|N|} \tag{4.4}$$

$$Recall_micro = \frac{\sum_c \frac{tp_c}{tp_c + fn_c}}{|N|} \tag{4.5}$$

$$Precision_macro = \frac{\sum_c Precision_c}{|N|} \tag{4.6}$$

$$Recall_macro = \frac{\sum_c Recall_c}{|N|} \tag{4.7}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.8}$$

$$F1 - score_macro = \frac{2 \times Precision_macro \times Recall_macro}{Precision_macro + Recall_macro} \tag{4.9}$$

$$F1 - score_micro = \frac{2 \times Precision_micro \times Recall_micro}{Precision_macro + Recall_micro} \tag{4.10}$$

$$FPR = \frac{fp}{tn + fp} \tag{4.11}$$

The performance of the stack ensemble learning approach employing various meta-learners is summarized in Table 4.3. The ensemble consists of base models including Random Forest (RF), Support Vector Classifier with radial basis function kernel (SVC rbf), Support Vector Classifier with linear kernel (SVC linear), Gaussian Naive Bayes, Logistic Regression, K-Nearest Neighbors (KNN), Gradient Boosting (GB), LightGBM (LGBM)[4], Decision Tree (DT) and XGBoost (XGB). Notably from Table 4.3 the ensemble achieves high performance across most metrics, with RF exhibiting the highest accuracy of 94.62 %, closely followed by SVC rbf and KNN. AUC scores also indicate robust model performance, with all models achieving scores above 0.95, highlighting the effectiveness of the ensemble approach in classification tasks. Notably, Decision Tree (DT) shows the lowest performance with an accuracy of 74.3% and F1 macro score of 81.36%. Random Forests are renowned for their ability to handle complex relationships in data, typically excelling in various prediction tasks. This effectiveness is evident in the context of predicting landslide triggers using the

TABLE 4.3: Performance of Stack ensemble learning with various meta learner

Metrics	RF	svc rbf	svc linear	gaussian NB	logistic reg	knn	gradient boosting	LGBM	DT	XG BOOST
accuracy	0.946	0.945	0.942	0.899	0.943	0.944	0.936	0.940	0.813	0.938
precision macro	0.945	0.943	0.942	0.914	0.941	0.943	0.935	0.937	0.86	0.936
precision micro	0.946	0.945	0.943	0.899	0.943	0.945	0.937	0.94	0.814	0.939
recall macro	0.946	0.945	0.942	0.898	0.943	0.944	0.937	0.94	0.814	0.938
recall micro	0.946	0.945	0.943	0.899	0.943	0.945	0.937	0.94	0.814	0.939
f1 macro	0.945	0.943	0.941	0.902	0.942	0.943	0.935	0.938	0.798	0.937
f1 micro	0.946	0.945	0.943	0.899	0.943	0.945	0.937	0.94	0.814	0.939
auc macro	0.996	0.995	0.995	0.953	0.996	0.979	0.991	0.995	0.95	0.994
auc micro	0.996	0.995	0.995	0.953	0.996	0.979	0.992	0.995	0.949	0.994

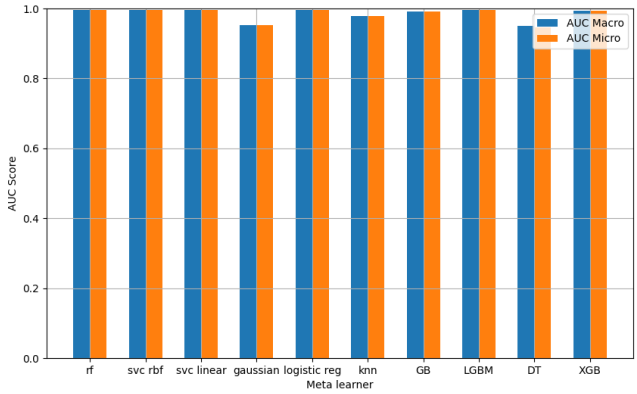


FIGURE 4.1: AUC score of various meta learners

stacking ensemble approach. Conversely, the decision tree performed poorly, with the lowest accuracy and F1 score. Decision trees may struggle on datasets with intricate relationships between features and target variables, especially when compared to models that are more adept at capturing complex patterns. Despite this, AUC scores also indicate robust model performance, with all models achieving scores above 0.95, highlighting the effectiveness of the ensemble approach in classification tasks

The AUC (Area Under the Curve) in Figure 4.1 shows the macro and micro auc score where AUC macro calculates the average AUC score for each class independently and then takes the mean of these scores. It evaluates model performance for each category separately, without considering class imbalances and AUC Micro aggregates true positives, false positives and false negatives across all classes to compute a single AUC score. It provides a unified measure, giving equal weight to each instance, irrespective of its class and is sensitive to class imbalances. Further support the robustness of RF, SVC RBF, LGBM and XGB with high values in both macro and

TABLE 4.4: Train and test time duration of different meta learner in mean ± sd format (sec)

Meta Learner	RF	SVC rbf	SVC linear	Gaussian	Logistic Reg
Train time	1920.43 ± 66.29	2197.03 ± 153.55	1859.60 ± 622.42	1746.31 ± 81.77	1734.93 ± 68.91
Test time	9.42 ± 0.68	11.72 ± 0.44	9.54 ± 0.99	8.69 ± 0.99	9.48 ± 0.69
Meta Learner	KNN	GB	LGMB	DT	XGB
Train time	1850.74 ± 78.62	2375.97 ± 470.82	1933.80 ± 43.48	1108.41 ± 15.91	1032.70 ± 38.43
Test time	14.07 ± 0.70	5.32 ± 0.22	10.27 ± 0.39	5.27 ± 0.26	5.30 ± 0.33

micro assessments. Conversely, decision tree registers comparatively lower AUC scores, indicating potential class imbalance or variability in performance across different classes.

In the context of landslide trigger prediction, minimizing false negatives (i.e., landslides that are not predicted when they occur) is crucial due to the severe consequences they can entail. Thus, the focus should be on models with high recall, as recall measures a model's ability to correctly identify positive instances (landslide triggers) out of all actual positive instances. In this regard, the Random Forest meta-learner demonstrated commendable recall values, making it a robust choice for this task. Normalized confusion matrices have been created for all the stacking ensemble models with different meta-learners, as shown in Figure 4.2. The term "normalized" implies that each of these groupings is represented as having 1.00 samples. By examining the confusion matrices for the various models, it is clear that activities labeled 2 and 9 are the most frequently confused with one another. Label 2 represents downpour, and label 9 represents rain, both of which are significant contributors to landslides. The similarity between these two activities leads to their confusion during classification. Further analysis reveals that class 2 and 9 are less confused with one another in the cases of the Random Forest and Logistic Regression meta-learners. This indicates that these models are better at distinguishing between similar triggers, enhancing their predictive accuracy. In contrast, the decision tree meta-learner exhibited the highest confusion between these labels, highlighting its limitation in handling nuanced distinctions within the data. This underscores the importance of selecting robust meta-learners like Random Forests to improve the performance of stacked ensemble models, particularly in critical applications such as landslide prediction.

The Table 4.4 provides insights into the training and testing time durations (in

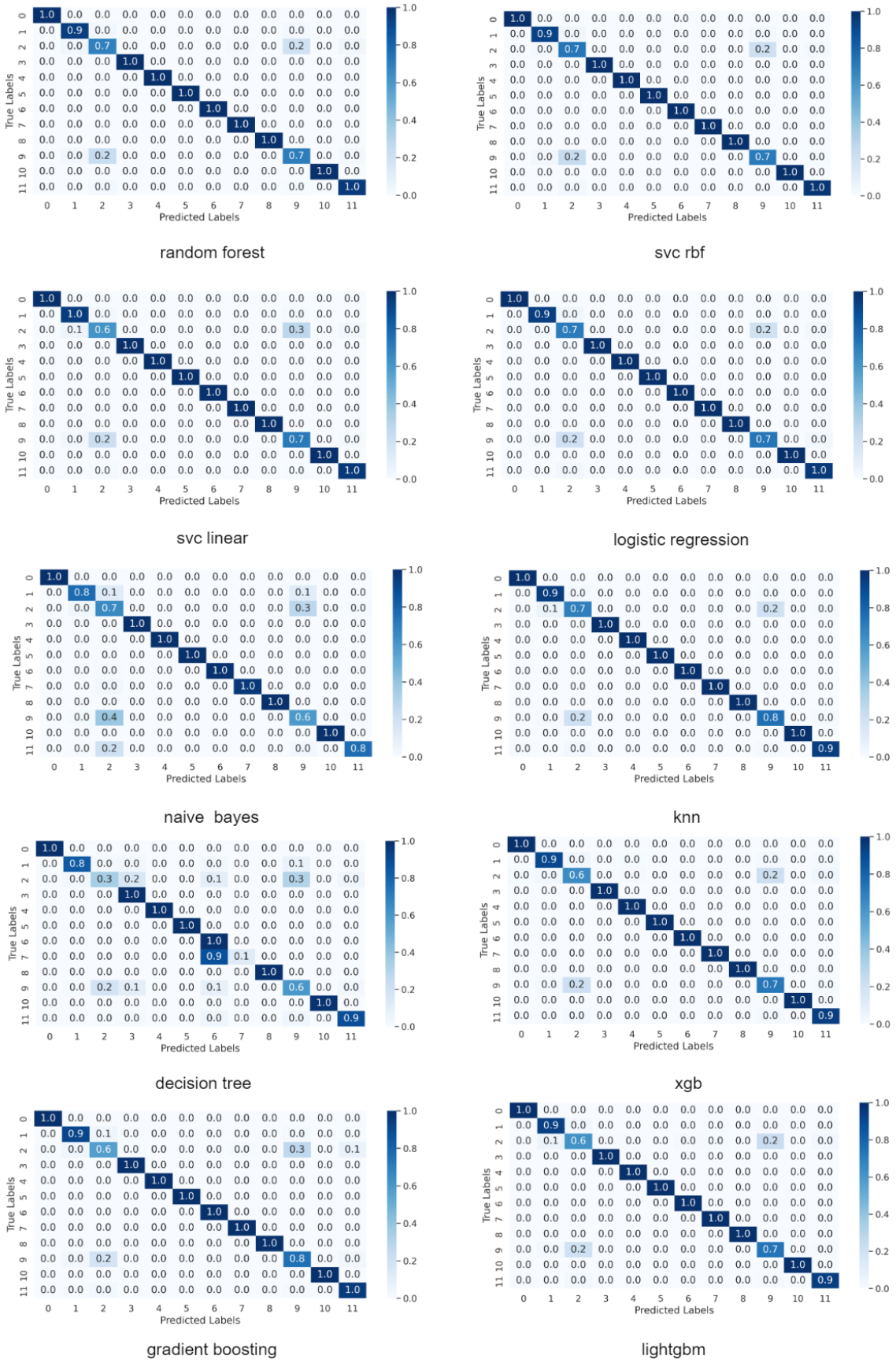


FIGURE 4.2: Confusion matrix for all meta learner in Stack ensemble learning

sec) of various meta-learners across different machine learning models. The mean and standard deviation (in $\pm$ format) are presented for each meta-learner and model combination. Gradient Boosting (GB) has the longest training time (2,375.97 ± 470.82 sec), while Decision Tree (DT) and XGBoost (XGB) are the quickest (1,108.41 ± 15.91 sec and 1,032.70 ± 38.43 sec, respectively). For testing times, K-Nearest Neighbors (KNN) is the slowest (14.07 ± 0.703 sec), whereas Decision Tree, XGBoost and Gradient Boosting are the fastest (around 5.300 sec each). Random Forest (RF) and SVC (rbf) exhibit relatively high training and testing times, indicating their computational intensity. This comparison helps in selecting meta-learners based on the desired balance between training duration and prediction efficiency. This information is valuable for assessing the computational efficiency of different meta-learners when applied to diverse machine learning models, aiding in the selection of an appropriate meta-learner based on time considerations for training and testing phases.

Chapter 5

Conclusion and Future Work

This chapter summarises landslide trigger prediction using our proposed stacked ensemble model and the challenges faced during this study. Future scope is also provided to show how this landslide trigger prediction can be improved and applied worldwide.

5.1 Conclusion

The main objective of this research is to accurately classify the triggers of landslides across India using various baseline machine learning (ML) models and deep learning models. By analyzing the predictions of multiple base models, such as Random Forest (RF), Gradient Boosting (GB), support vector machine (SVC), k-nearest neighbor (KNN), logistic regression (LR) and naive Bayes (NB), the ensemble model demonstrates improved accuracy, precision, recall, F1 score and AUC compared to individual models. The stacking ensemble approach, incorporating 16 MLPs with RF as the meta-learner, proves to be a promising technique for landslide trigger classification, achieving an overall accuracy of 94.62%. This surpasses the performance of other baseline models, including different types of meta-learners. The stacking ensemble effectively captures the complex relationships between various factors contributing to landslide triggers. Its ability to generalize well to unseen data and its robust performance across different evaluation metrics highlight its potential for real-world applications in landslide trigger classification. Additionally, the flexibility of the stacking ensemble allows for the incorporation of diverse base models, enabling the exploitation of various modeling techniques and data characteristics.

Creating the datasets for this study was the most challenging part. A thorough investigation of the factors causing landslides was conducted through a literature review. After combining data from several sources, an imbalance in the dataset was identified. The machine learning models did not yield good results with this unbalanced dataset. To address this issue, the oversampling approach (SMOTE) was employed to create a balanced dataset, leading to significantly better outcomes.

5.2 Future Work

Mass movements such as landslides are a major natural hazard in mountainous regions all over the world [60]. Annually, landslides cause 1000 deaths and about USD 4 billion loss of property worldwide [61]. However in our study we considered the landslide location of India only. In later study this research could be applied to other landslide-prone areas around the world. However, in this study only landslide triggers are considered. But this study does not go into detail about the specific impact of each trigger on casualties and damages. Also, landslide category factors such as complex, debris flow, mudslide can be analysed for prior detection of landslides. This study does not make use of an image dataset. In the following study, the consequences of landslides can be examined by analyzing photos taken before and after the events.

Bibliography

[1] M Ercanoglu, O Kasmer, and N Temiz. "Adaptation and comparison of expert opinion to analytical hierarchy process for landslide susceptibility mapping". In: *Bulletin of Engineering Geology and the Environment* 67 (2008), pp. 565–578.

[2] Fausto Guzzetti et al. "Landslide hazard evaluation: a review of current techniques and their application in a multi-scale study, Central Italy". In: *Geomorphology* 31.1-4 (1999), pp. 181–216.

[3] Yashar Alimohammadlou, Asadallah Najafi, and Ali Yalcin. "Landslide process and impacts: A proposed classification method". In: *Catena* 104 (2013), pp. 219–232.

[4] Vinod Chandra SS and E Shaji. "Landslide identification using machine learning techniques: Review, motivation, and future prospects". In: *Earth science informatics* 15.4 (2022), pp. 2063–2090.

[5] Malcolm G Anderson and Elizabeth Holcombe. *Community-based landslide risk reduction: managing disasters in small steps*. World Bank Publications, 2013.

[6] Debarati Guha-Sapir et al. "Annual disaster statistical review 2010". In: *Centre for Research on the Epidemiology of Disasters* (2011), pp. 1–80.

[7] Harley Betts et al. "Development of a landslide component for a sediment budget model". In: *Environmental modelling & software* 92 (2017), pp. 28–39.

[8] Hamid Reza Pourghasemi and Omid Rahmati. "Prediction of the landslide susceptibility: Which algorithm, which precision?" In: *Catena* 162 (2018), pp. 177–192.

[9] Samuel T. McColl. "Landslide causes and triggers". In: *Landslide Hazards, Risks, and Disasters*. Elsevier, 2022, pp. 13–41. DOI: 10.1016/b978-0-12-818464-6.00011-1. URL: https://doi.org/10.1016/b978-0-12-818464-6.00011-1.

[10] Abhirup Dikshit et al. "Rainfall Induced Landslide Studies in Indian Himalayan Region: A Critical Review". In: *Applied Sciences* 10.7 (Apr. 2020), p. 2466. DOI: 10.3390/app10072466. URL: https://doi.org/10.3390/app10072466.

[11] Abhirup Dikshit et al. "Rainfall induced landslide studies in Indian Himalayan region: a critical review". In: *Applied Sciences* 10.7 (2020), p. 2466.

[12] Zhilu Chang et al. "Landslide susceptibility prediction using slope unit-based machine learning models considering the heterogeneity of conditioning factors". In: *Journal of Rock Mechanics and Geotechnical Engineering* 15.5 (2023), pp. 1127–1143.

[13] Zhongqiang Liu et al. "Modelling of shallow landslides with machine learning algorithms". In: *Geoscience Frontiers* 12.1 (Jan. 2021), pp. 385–393. DOI: 10.1016/j.gsf.2020.04.014. URL: https://doi.org/10.1016/j.gsf.2020.04.014.

[14] C. W. W. Ng et al. "Spatiotemporal modelling of rainfall-induced landslides using machine learning". In: *Landslides* 18.7 (Apr. 2021), pp. 2499–2514. DOI: 10.1007/s10346-021-01662-0. URL: https://doi.org/10.1007/s10346-021-01662-0.

[15] Haojie Wang et al. "Landslide identification using machine learning". In: *Geoscience Frontiers* 12.1 (Jan. 2021), pp. 351–364. DOI: 10.1016/j.gsf.2020.02.012. URL: https://doi.org/10.1016/j.gsf.2020.02.012.

[16] Defang Liu, Junjie Li, and Fenglei Fan. "Recognition of landslide triggers in southeast Tibetan (China) using a novel lightweight network". In: *Environmental Earth Sciences* 81.8 (2022), p. 234.

[17] Quinton Aguilera et al. "On the prediction of landslide occurrences and sizes via Hierarchical Neural Networks". In: *Stochastic Environmental Research and Risk Assessment* 36.8 (Apr. 2022), pp. 2031–2048. DOI: 10.1007/s00477-022-02215-0. URL: https://doi.org/10.1007/s00477-022-02215-0.

[18] S. Sreelakshmi and S. S. Vinod Chandra. "Landslide Classification Using Deep Convolutional Neural Network with Synthetic Minority Oversampling Technique". In: *Lecture Notes in Computer Science*. Springer Nature Switzerland,

2023, pp. 240–252. DOI: 10.1007/978-3-031-24848-1_17. URL: https://doi.org/10.1007/978-3-031-24848-1_17.

[19] Trong-An Bui et al. "Deep learning for landslide recognition in satellite architecture". In: *IEEE Access* 8 (2020), pp. 143665–143678.

[20] Hossein Moayedi et al. "Modification of landslide susceptibility mapping using optimized PSO-ANN technique". In: *Engineering with Computers* 35.3 (Sept. 2018), pp. 967–984. DOI: 10.1007/s00366-018-0644-0. URL: https://doi.org/10.1007/s00366-018-0644-0.

[21] Jie Dou et al. "Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan". In: *Landslides* 17 (2020), pp. 641–658.

[22] Poonam Kainthura and Neelam Sharma. "Hybrid machine learning approach for landslide prediction, Uttarakhand, India". In: *Scientific Reports* 12.1 (Nov. 2022). DOI: 10.1038/s41598-022-22814-9. URL: https://doi.org/10.1038/s41598-022-22814-9.

[23] Dieu Tien Bui et al. "Shallow Landslide Prediction Using a Novel Hybrid Functional Machine Learning Algorithm". In: *Remote Sensing* 11.8 (Apr. 2019), p. 931. DOI: 10.3390/rs11080931. URL: https://doi.org/10.3390/rs11080931.

[24] Naruephorn Tengtrairat et al. "Automated Landslide-Risk Prediction Using Web GIS and Machine Learning Models". In: *Sensors* 21.13 (July 2021), p. 4620. DOI: 10.3390/s21134620. URL: https://doi.org/10.3390/s21134620.

[25] Mohammed Sarfaraz Gani Adnan et al. "The potential of Tidal River Management for flood alleviation in South Western Bangladesh". In: *Science of The Total Environment* 731 (Aug. 2020), p. 138747. DOI: 10.1016/j.scitotenv.2020.138747. URL: https://doi.org/10.1016/j.scitotenv.2020.138747.

[26] Zhilu Chang et al. "Landslide susceptibility prediction using slope unit-based machine learning models considering the heterogeneity of conditioning factors". In: *Journal of Rock Mechanics and Geotechnical Engineering* (Aug. 2022). DOI: 10.1016/j.jrmge.2022.07.009. URL: https://doi.org/10.1016/j.jrmge.2022.07.009.

[27] Mohammad Azarafza et al. "Deep learning-based landslide susceptibility mapping". In: *Scientific Reports* 11.1 (Dec. 2021). DOI: 10.1038/s41598-021-03585-1. URL: https://doi.org/10.1038/s41598-021-03585-1.

[28] Charles Wang Wai Ng et al. "Spatiotemporal modelling of rainfall-induced landslides using machine learning". In: *Landslides* 18.7 (2021), pp. 2499–2514.

[29] Pranav Rajendra Rane and Shweta Vincent. "Landslide susceptibility mapping using machine learning algorithms for Nainital, India". In: *Engineered Science* 17 (2021), pp. 142–155.

[30] Lingfeng He et al. "Maximizing Impacts of Remote Sensing Surveys in Slope Stability—A Novel Method to Incorporate Discontinuities into Machine Learning Landslide Prediction". In: *ISPRS International Journal of Geo-Information* 10.4 (Apr. 2021), p. 232. DOI: 10.3390/ijgi10040232. URL: https://doi.org/10.3390/ijgi10040232.

[31] Yong gang Zhang et al. "Application of an enhanced BP neural network model with water cycle algorithm on landslide prediction". In: *Stochastic Environmental Research and Risk Assessment* 35.6 (Oct. 2020), pp. 1273–1291. DOI: 10.1007/s00477-020-01920-y. URL: https://doi.org/10.1007/s00477-020-01920-y.

[32] Yacine Achour and Hamid Reza Pourghasemi. "How do machine learning techniques help in increasing accuracy of landslide susceptibility maps?" In: *Geoscience Frontiers* 11.3 (May 2020), pp. 871–883. DOI: 10.1016/j.gsf.2019.10.001. URL: https://doi.org/10.1016/j.gsf.2019.10.001.

[33] Yaning Yi and Wanchang Zhang. "A new deep-learning-based approach for earthquake-triggered landslide detection from single-temporal RapidEye satellite imagery". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 6166–6176.

[34] Viet-Ha Nhu et al. "Landslide Susceptibility Mapping Using Machine Learning Algorithms and Remote Sensing Data in a Tropical Environment". In: *International Journal of Environmental Research and Public Health* 17.14 (July 2020), p. 4933. DOI: 10.3390/ijerph17144933. URL: https://doi.org/10.3390/ijerph17144933.

[35] Martin Kuradusenge, Santhi Kumaran, and Marco Zennaro. "Rainfall-Induced Landslide Prediction Using Machine Learning Models: The Case of Ngororero District, Rwanda". In: *International Journal of Environmental Research and Public Health* 17.11 (June 2020), p. 4147. DOI: 10.3390/ijerph17114147. URL: https://doi.org/10.3390/ijerph17114147.

[36] Dong Van Dao et al. "A spatially explicit deep learning neural network model for the prediction of landslide susceptibility". In: *CATENA* 188 (May 2020), p. 104451. DOI: 10.1016/j.catena.2019.104451. URL: https://doi.org/10.1016/j.catena.2019.104451.

[37] Ali Mutar Fanos and Biswajeet Pradhan. "A novel hybrid machine learning-based model for rockfall source identification in presence of other landslide types using LiDAR and GIS". In: *Earth Systems and Environment* 3 (2019), pp. 491–506.

[38] Pham et al. "Landslide Susceptibility Assessment by Novel Hybrid Machine Learning Algorithms". In: *Sustainability* 11.16 (Aug. 2019), p. 4386. DOI: 10.3390/su11164386. URL: https://doi.org/10.3390/su11164386.

[39] Prima Kadavi, Chang-Wook Lee, and Saro Lee. "Application of Ensemble-Based Machine Learning Models to Landslide Susceptibility Mapping". In: *Remote Sensing* 10.8 (Aug. 2018), p. 1252. DOI: 10.3390/rs10081252. URL: https://doi.org/10.3390/rs10081252.

[40] Caroline S. Juang, Thomas A. Stanley, and Dalia B. Kirschbaum. "Using citizen science to expand the global map of landslides: Introducing the Cooperative Open Online Landslide Repository (COOLR)". In: *PLOS ONE* 14.7 (July 2019). Ed. by Christian Huggel, e0218657. DOI: 10.1371/journal.pone.0218657. URL: https://doi.org/10.1371/journal.pone.0218657.

[41] Noel Gorelick et al. "Google Earth Engine: Planetary-scale geospatial analysis for everyone". In: *Remote Sensing of Environment* 202 (Dec. 2017), pp. 18–27. DOI: 10.1016/j.rse.2017.06.031. URL: https://doi.org/10.1016/j.rse.2017.06.031.

[42] Yumiao Wang et al. "Optimizing the predictive ability of machine learning methods for landslide susceptibility mapping using SMOTE for Lishui City in Zhejiang Province, China". In: *International journal of environmental research and public health* 16.3 (2019), p. 368.

[43] Thomas G. Dietterich. "Ensemble Methods in Machine Learning". In: *Multiple Classifier Systems*. Springer Berlin Heidelberg, 2000, pp. 1–15. DOI: 10.1007/3-540-45014-9_1. URL: https://doi.org/10.1007/3-540-45014-9_1.

[44] Yuzhen Zhang, Jingjing Liu, and Wenjuan Shen. "A Review of Ensemble Learning Algorithms Used in Remote Sensing Applications". In: *Applied Sciences* 12.17 (Aug. 2022), p. 8654. DOI: 10.3390/app12178654. URL: https://doi.org/10.3390/app12178654.

[45] Leo Breiman. "Bagging predictors". In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140. DOI: 10.1007/bf00058655. URL: https://doi.org/10.1007/bf00058655.

[46] Artur J. Ferreira and Mário A. T. Figueiredo. "Boosting Algorithms: A Review of Methods, Theory, and Applications". In: *Ensemble Machine Learning*. Springer US, 2012, pp. 35–85. DOI: 10.1007/978-1-4419-9326-7_2. URL: https://doi.org/10.1007/978-1-4419-9326-7_2.

[47] Bohdan Pavlyshenko. "Using Stacking Approaches for Machine Learning Models". In: *2018 IEEE second international conference on data stream mining & processing (DSMP)*. IEEE, Aug. 2018. DOI: 10.1109/dsmp.2018.8478522. URL: https://doi.org/10.1109/dsmp.2018.8478522.

[48] Alberto Fernandez et al. "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary". In: *Journal of Artificial Intelligence Research* 61 (Apr. 2018), pp. 863–905. DOI: 10.1613/jair.1.11192. URL: https://doi.org/10.1613/jair.1.11192.

[49] Leo Breiman. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/a:1010933404324. URL: https://doi.org/10.1023/a:1010933404324.

[50] Mohebbanaaz, L. V. Rajani Kumari, and Y. Padma Sai. "Classification of ECG beats using optimized decision tree and adaptive boosted optimized decision tree". In: *Signal, Image and Video Processing* 16.3 (Oct. 2021), pp. 695–703. DOI:

10.1007/s11760-021-02009-x. URL: https://doi.org/10.1007/s11760-021-02009-x.

[51] William S Noble. "What is a support vector machine?" In: *Nature Biotechnology* 24.12 (Dec. 2006), pp. 1565–1567. DOI: 10.1038/nbt1206-1565. URL: https://doi.org/10.1038/nbt1206-1565.

[52] Kashvi Taunk et al. "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification". In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, May 2019. DOI: 10.1109/iccs45141.2019.9065747. URL: https://doi.org/10.1109/iccs45141.2019.9065747.

[53] Alexey Natekin and Alois Knoll. "Gradient boosting machines, a tutorial". In: *Frontiers in Neurorobotics* 7 (2013). DOI: 10.3389/fnbot.2013.00021. URL: https://doi.org/10.3389/fnbot.2013.00021.

[54] Alfred DeMaris. "A Tutorial in Logistic Regression". In: *Journal of Marriage and the Family* 57.4 (Nov. 1995), p. 956. DOI: 10.2307/353415. URL: https://doi.org/10.2307/353415.

[55] Irina Rish et al. "An empirical study of the naive Bayes classifier". In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.

[56] Tianqi Chen and Carlos Guestrin. "XGBoost". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Aug. 2016. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[57] Zhongqiang Liu et al. "Modelling of shallow landslides with machine learning algorithms". In: *Geoscience Frontiers* 12.1 (2021), pp. 385–393.

[58] Haojie Wang et al. "Landslide identification using machine learning". In: *Geoscience Frontiers* 12.1 (2021), pp. 351–364.

[59] Mohammed Sarfaraz Gani Adnan et al. "Improving spatial agreement in machine learning-based landslide susceptibility mapping". In: *Remote Sensing* 12.20 (2020), p. 3347.

[60] Hamid Reza Pourghasemi et al. "Assessment of landslide-prone areas and their zonation using logistic regression, logitboost, and naïvebayes machine-learning algorithms". In: *Sustainability* 10.10 (2018), p. 3697.

[61] Saro Lee and Biswajeet Pradhan. "Probabilistic landslide hazards and risk mapping on Penang Island, Malaysia". In: *Journal of Earth System Science* 115 (2006), pp. 661–672.