Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.6, December 2025, pp. 3853~3861
e-ISSN: 2548-6861

Stacking of DT, RF, and Gradient Boosting Algorithms for Classification of Building Damage Due to Earthquakes

Nur Aqliah Ilmi [1]*, Nurul A. S Winarsih [2]*
Teknik Informatika, Universitas Dian Nuswantoro
111202214039@mhs.dinus.ac.id [1],nurulanisasw@dsn.dinus.ac.id [2*]

Article Info

Article history:

Received 2025-09-19
Revised 2025-11-16
Accepted 2025-12-10

Keyword:

Building Damage,
Earthquake Clasification,
Ensemble Stacking,
ADASYN.

ABSTRACT

Classification of building damage levels due to earthquakes is an important aspect in disaster mitigation and post-disaster risk assessment. This study aims to improve classification accuracy on imbalanced data using an ensemble stacking method. It combines Decision Tree, Random Forest, and Gradient Boosting algorithms, with Logistic Regression as a meta-learner. The building damage dataset from the 2015 Gorkha Nepal earthquake underwent data cleaning, categorical transformation, normalization, and balancing using ADASYN. Evaluation showed that Random Forest was the best single model. The stacking model achieved the highest accuracy of 91.77% after balancing. These results show that stacking improves generalization and classification accuracy on imbalanced data. This suggests significant potential for integration into disaster decision-support systems that require fast, accurate building-damage assessment.

I. INTRODUCTION

Indonesia is highly vulnerable to natural disasters, especially earthquakes. This is due to its geographical position, which is located in the meeting zone of three major tectonic plates, the Indo-Australian, Pacific, and Eurasian Plates, and is located on the Pacific Ring of Fire, known as the region with the most active seismic activity in the world [1]. As a result, earthquakes often cause building damage, loss of life, and socio-economic disruption. In this context, classifying the level of building damage from earthquakes is an important step in government and related institutions' disaster mitigation efforts and emergency response policy planning [2].

In this context, classifying the extent of earthquake-induced building damage is crucial for supporting disaster mitigation policies, emergency response planning, and post-disaster damage assessment. An accurate classification model can help disaster agencies expedite building damage assessment, optimize resource allocation, and serve as an integral component of a data-driven early warning system.

However, the main problem often encountered in building damage classification is data imbalance, where the number of buildings with light, moderate, and severe damage levels is disproportionate. This condition causes machine learning models to be biased towards the majority class and less able to recognize patterns in the minority class effectively. Several previous studies have attempted to overcome this problem by using feature selection techniques based on Particle Swarm Optimization (PSO) and applying classical algorithms such as K-Nearest Neighbors, Decision Trees, and Random Forests [3]. However, these approaches still have limitations in terms of generalization ability and prediction accuracy in the minority class.

This study proposes the use of a stacking-type ensemble learning method, which combines several basic algorithms, namely Decision Tree, Random Forest, and Gradient Boosting, with Logistic Regression as a meta-learner. The stacking ensemble approach was chosen because it is flexible in combining various types of models and is effective in increasing accuracy and generalization capabilities compared to homogeneous ensemble methods such as bagging and boosting [4][5].

Furthermore, this study integrates Adaptive Synthetic Sampling (ADASYN)-based data-balancing techniques to address class imbalance. Therefore, this study aims to develop an accurate, stable, and adaptive earthquake damage

classification model that addresses data imbalance, thereby further contributing to rapid, data-driven decision-support systems for disaster mitigation, early warning, and post-disaster evaluation.

II. METODE

This study utilizes a machine learning approach by applying three main algorithms, namely Decision Tree, Random Forest, and Gradient Boosting, which are then combined in an ensemble stacking model to obtain more optimal prediction performance by combining the advantages of each base model. The ensemble stacking approach was chosen because it can reduce the weaknesses of individual algorithms while increasing the stability, generalization, and predictive accuracy. The selection of the combination of the three algorithms, namely Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB), in this study is based on their structural characteristics, advantages, and differences in approaches to bias and variance. These three algorithms have complementary strengths in handling diverse and complex data, so when combined in a stacking ensemble framework, the resulting model becomes more accurate, stable, and better able to generalize. The dataset used is global data previously processed by Ghimire et al. (2022) and derived from field data on the impact of the Gorkha earthquake that hit the Gandaki Pradesh region of Nepal in April 2015 with a magnitude of 7.8 Mw. This earthquake was recorded as one of the most devastating disasters ever to occur in Nepal, causing economic losses estimated at 10 billion US dollars or almost half of Nepal's nominal Gross Domestic Product (GDP) at that time, and resulting in more than 9,000 fatalities, hundreds of thousands of injuries, and millions of others losing their homes, jobs, and sources of livelihood [16].

The data used in this study were collected through a large-scale survey conducted by Kathmandu Living Labs in collaboration with the Nepal Central Bureau of Statistics, making it one of the largest and most comprehensive post-disaster datasets ever compiled worldwide. The dataset contains detailed information on the extent of building damage, the structural condition of dwellings, the impact on public infrastructure, and social, economic, and demographic data on communities directly and indirectly affected by the disaster. Overall, the dataset comprises 40 features and 260,601 entries [6], providing a valuable data source for developing building damage prediction models.

One of these features is the id or building_id column, which serves only as a unique identifier for each dataset entry and contains no predictive information about the target variable. Because its presence does not contribute to the model's learning and only adds complexity to the data, this column was removed during data preprocessing to ensure the model focuses on relevant and informative features. After removing this column, the number of features used in model training was reduced to 39. This step was taken to simplify the data structure, reduce the risk of overfitting, speed up training time, and support the improvement of the efficiency and accuracy of the building damage prediction model developed in this study

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 260601 entries, 0 to 260600
Data columns (total 39 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   geo_level_1_id                          260601 non-null  int64
 1   geo_level_2_id                          260601 non-null  int64
 2   geo_level_3_id                          260601 non-null  int64
 3   count_floors_pre_eq                     260601 non-null  int64
 4   age                                     260601 non-null  int64
 5   area_percentage                         260601 non-null  int64
 6   height_percentage                       260601 non-null  int64
 7   land_surface_condition                  260601 non-null  object
 8   foundation_type                         260601 non-null  object
 9   roof_type                               260601 non-null  object
 10  ground_floor_type                       260601 non-null  object
 11  other_floor_type                        260601 non-null  object
 12  position                                260601 non-null  object
 13  plan_configuration                      260601 non-null  object
 14  has_superstructure_adobe_mud            260601 non-null  int64
 15  has_superstructure_mud_mortar_stone     260601 non-null  int64
 16  has_superstructure_stone_flag           260601 non-null  int64
 17  has_superstructure_cement_mortar_stone  260601 non-null  int64
 18  has_superstructure_mud_mortar_brick     260601 non-null  int64
 19  has_superstructure_cement_mortar_brick  260601 non-null  int64
 20  has_superstructure_timber               260601 non-null  int64
 21  has_superstructure_bamboo               260601 non-null  int64
 22  has_superstructure_rc_non_engineered    260601 non-null  int64
 23  has_superstructure_rc_engineered        260601 non-null  int64
 24  has_superstructure_other                260601 non-null  int64
 25  legal_ownership_status                  260601 non-null  object
 26  count_families                          260601 non-null  int64
 27  has_secondary_use                       260601 non-null  int64
 28  has_secondary_use_agriculture           260601 non-null  int64
 29  has_secondary_use_hotel                 260601 non-null  int64
 30  has_secondary_use_rental                260601 non-null  int64
 31  has_secondary_use_institution           260601 non-null  int64
 32  has_secondary_use_school                260601 non-null  int64
 33  has_secondary_use_industry              260601 non-null  int64
 34  has_secondary_use_health_post           260601 non-null  int64
 35  has_secondary_use_gov_office            260601 non-null  int64
 36  has_secondary_use_use_police            260601 non-null  int64
 37  has_secondary_use_other                 260601 non-null  int64
 38  damage_grade                            260601 non-null  int64
```

Figure 1. Data Reduction Results

Before being used in the modeling process, data can undergo a series of preprocessing steps, systematically arranged to ensure consistency, cleanliness, and appropriateness. The first step used is the data cleaning stage, which aims to identify and correct entries containing missing values, redundant information, or format inconsistencies [7]. Handling of empty values is not done haphazardly; the approach used considers the context of the variables and the distribution of the data to avoid distorting the overall statistical structure. After cleaning, a transformation is applied to categorical features, which are non-numeric variables that machine learning algorithms cannot directly process. To overcome this, the one-hot encoding technique is used, namely a representation method in which each category is converted into a binary vector, allowing the model to compare classes without imposing an irrelevant numeric order [8]. The next step is to normalize the contributing features, standardizing their values to a uniform scale so that each variable has a proportional contribution to model training. The next step is to apply ADASYN (Adaptive Synthetic Sampling), an oversampling technique that addresses class imbalance by adaptively generating additional synthetic samples for the minority class. This process involves giving greater weight to minority data that is more difficult for machine learning models to classify, thereby achieving a more balanced data distribution and improved classification performance [18].

Furthermore, to ensure reliable training results and avoid data-splitting bias, the data were stratified, keeping the

proportions of each damage class consistent between the training and test sets. The dataset was split into two parts at a 70:30 ratio, with 70% used for training and 30% for testing. This stratification approach is important for maintaining class balance within each data subset. Furthermore, to assess the model's stability and generalization, this study used k-fold cross-validation with k = 5, dividing the dataset into five equally sized subsets. At each iteration, four subsets were used for training, and the remaining subsets were used for testing in turn. The average value of all iterations was used as a measure of the model's final performance. This approach ensures that evaluation results are not dependent on a single data split and provides a more accurate picture of the model's stability and consistency in classifying the level of earthquake-induced building damage.

Conceptually, a Decision Tree is a hierarchical, rule-based learning model that is relatively easy to interpret and efficient at finding relationships among variables, without assuming linearity. However, this model has high variance, making it sensitive to small changes in the training data. To overcome this weakness, Random Forest is used as a bagging method that combines many decision trees trained on random subsets of data and features. This approach is effective in reducing variance and improving prediction stability because the final result is obtained by averaging or majority voting across multiple Decision trees [10] [11].

Unlike Random Forest, Gradient Boosting works on the principle of boosting, which is building a model gradually, where each new model learns from the errors (residuals) of the previous model. This makes Gradient Boosting exhibit low bias and strong ability to capture non-linear patterns and complex feature interactions [12] [13]. However, this iterative nature can also increase the risk of overfitting if regularization or early stopping is not performed.

The hyperparameter tuning stage is a crucial part of machine learning model development because it determines the parameter configuration that produces optimal performance. Hyperparameters are parameters whose values are not learned directly from the data but are determined before the training process begins, such as the number of estimators, tree depth, or learning rate. Selecting the correct hyperparameter values can affect the balance between model bias and variance, thereby impacting the level of accuracy and the generalization ability to new data [19].

Stacking is an ensemble learning method that combines predictions from several base models (base learners) via a meta-learner to produce a more accurate final prediction. This method is designed to improve generalization capabilities by leveraging the strengths of various machine learning algorithms. This method consists of two levels of learning, namely Level-0 (base learners) and Level-1 (meta learner). At Level-0, several models, such as Decision Trees, Random Forests, and Gradient Boosting, are trained in parallel to produce initial predictions. The predictions generated by these base models are then used as input features for the meta-learner at Level-1. This meta-model, usually implemented as Logistic Regression or other simpler algorithms, is tasked with learning the relationships between predictions and producing a more accurate final decision [12]. This approach allows stacking to combine the advantages of various algorithms while minimizing their respective weaknesses, thereby improving the overall performance of the predictive system [15].

The combination of these three algorithms in the stacking ensemble model is based on the bias–variance tradeoff principle, in which each algorithm contributes to balancing the strengths and weaknesses of the others. Decision Tree provides speed and interpretability; Random Forest maintains stability through bagging; and Gradient Boosting corrects residual errors through boosting. Thus, the stacking model can achieve better performance than a single approach [16].

As a meta-learner, Logistic Regression is used to study the relationship between predictions from the three base models at Level-1 of learning. Logistic Regression optimizes the contribution weights of each base learner based on their performance on the training data, resulting in more accurate final predictions and stronger generalization. This approach is practical in various ensemble learning studies, especially for the classification of imbalanced data in the fields of disaster and health [5].

With this strategy, stacking leverages the strengths of each algorithm to reduce classification errors, increase prediction stability, and improve model generalization to complex and imbalanced data. This approach makes the combination of DT, RF, and GB relevant for classifying earthquake-induced building damage levels, where class distributions across damage categories are often uneven and complex.

Model evaluation in this study aims to assess the model's ability to classify class labels on previously unseen data accurately. Four evaluation metrics are used: accuracy, precision, recall, and F1-score. Accuracy is the proportion of correct predictions (both positive and negative) among the total number of predictions made by the model. This metric provides an overview of how often the model produces predictions that match reality [13].

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Precision in machine learning is a metric that measures the proportion of optimistic predictions that are actually positive.

$$Precission = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

F1-score is a harmonic mean between precision and recall, which is designed to provide a balance between the two metrics, especially when there is a trade-off between the two [14].

$$F1 - score = 2\mathrm{x}\frac{\mathrm{Recall\ x\ Precission}}{\mathrm{Recall} + \mathrm{Precission}}$$

Recall is a metric that measures the proportion of actual positives the model correctly predicts.

$$Recall = \frac{TP}{\text{TP} + \text{FN}}$$

All these metrics are calculated for each class and then averaged using a macroaverage method to assign equal weight to all classes. In addition to numerical evaluation, a confusion matrix is used as a visual representation of classification errors between classes. This matrix provides a clearer view of the distribution of correct and incorrect predictions, helping evaluate the model's strengths and weaknesses relative to the data structure.

III. RESULTS AND DISCUSSION

This chapter presents a comparison of the Decision Tree, Random Forest, and Gradient Boosting algorithms after and after the data balancing process using the Adaptive Synthetic Sampling Technique (ADASYN, and compared with the results of research conducted by Winarsih et al. [3] who applied the Particle Swarm Optimization (PSO) based feature selection method to the Decision Tree and Random Forest algorithms. The results of et al.'s research show that the Decision Tree algorithm's accuracy increased significantly from 65.7% to 72.7%, indicating that removing irrelevant features makes the model more focused and effective at building classification patterns. Meanwhile, Random Forest remains the most superior model with the Phasor PSO variant. The minor increase in Random Forest compared to Decision Tree indicates that Random Forest has remained relatively stable from the start in handling many features, but combining it with PSO still provides improved performance while maintaining stable prediction results. Furthermore, in this study, data balancing was performed to address the uneven class distribution in earthquake-induced building damage data, thereby reducing the model's bias towards specific classes. This evaluation aims to assess the effectiveness of each model in classifying earthquake-caused building damage into three categories: Minor Damage, Moderate Damage, and Severe Damage. Data balancing may be performed first because the initial dataset is highly skewed toward the majority class, potentially affecting the model's ability to identify minority classes accurately. The following is a comparison table between machine learning models before and after using ADASYN:

TABLE I
DECISION TREE BEFORE ADASYN

Damage	Accuracy	Precission	F1-Score	Recall
Light	66.11	49.83	51.01	52.23
Moderate	66.11	71.76	71.01	70.52
Severe	66.11	61.64	62.12	62.60

TABLE II
RANDOM FOREST BEFORE ADASYN

Damage	Accuracy	Precission	F1-Score	Recall
Light	72.07	64.80	55.17	48.03
Moderate	72.07	72.93	77.34	82.32
Severe	72.07	71.95	66.35	61.56

TABEL III
GRADIEN BOOSTING BEFORE ADASYN

Damage	Accuracy	Precission	F1-Score	Recall
Light	68.35	63.46	46.74	36.99
Moderate	68.35	67.71	75.62	85.64
Severe	68.35	71.63	57.47	47.98

Before data balancing, the three Decision Tree, Random Forest, and Gradient Boosting models showed varying results. Decision Tree had an accuracy of 72.07% across all classes, with relatively good performance in the moderate damage class (F1-score 71.01%), but still lower in the light (F1-score 51.01%) and heavy (F1-score 62.12%) classes. Random Forest showed an overall accuracy of 72.07% with the best performance in the moderate damage class (F1-score 77.34) and the highest recall of 82.32%, indicating its ability to better recognize moderate damage compared to other classes. Meanwhile, Gradient Boosting recorded an accuracy of 68.35% with the highest recall of 85.64% in the moderate class, but its performance in the light damage class was relatively low (F1-score 46.74%). In general, all three methods tended to be better at classifying moderate damage, while light damage was the most challenging category to recognize. Of the three algorithms, Random Forest has the most balanced performance before data balancing is applied.

From the initial distribution of damage in the dataset, it shows a significant imbalance, where Moderate Damage dominates the data with a total of 148,259 samples, followed by Major Damage with 87,128 and finally Minor Damage with only 25,124.

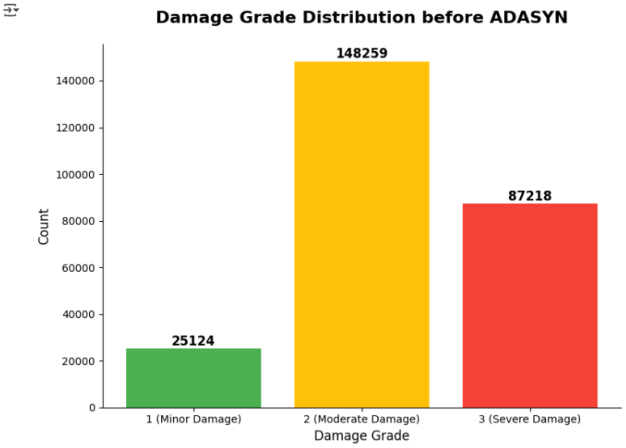

Figure 2. Distribution of Damage Grade Before ADASYN

To overcome this problem, the Adaptive Synthetic (ADASYN) method is used for data balancing. ADASYN generates synthetic samples for minority classes by sampling from the distributions of their nearest neighbors, which are difficult to predict. The results of the ADASYN data balancing approach are shown in the figure below, which shows that the distribution of the Light Damage class is superior to the other damage classes: 149,982 data points. In comparison, the Moderate Damage class has 148,259 data points, and the Severe Damage class has 140,828 data points. This shows that the machine learning model for the Light Damage class has the highest value among the other damage classes.

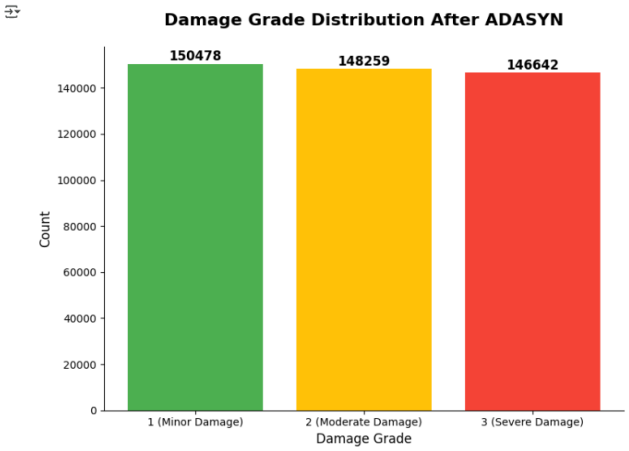

Figure 3. Distribution of Damage Grade After ADASYN

The distribution of data before and after the balancing process is an important factor in assessing the effectiveness of the Adaptive Synthetic Sampling (ADASYN) method in this study. The dataset used consists of 260,601 building data entries with three damage levels: light, moderate, and severe. Before the balancing process, the data showed significant imbalances: 25,124 samples in the light-damage class, 148,259 in the moderate-damage class, and 87,128 in the severe-damage class. This imbalanced distribution can bias the machine learning model toward the majority class (moderate damage) and lead to poor performance in recognizing patterns in the minority classes (light and severe damage). To overcome this problem, the ADASYN method was applied to enrich the representation of the minority class by generating synthetic data based on the difficult-to-predict nearest-neighbor distribution. After the balancing process, the data distribution became more balanced, with 149,982 samples for the light damage class, 148,259 for the moderate damage class, and 140,828 for the severe damage class. This change indicates that the ADASYN method has successfully reduced class imbalance, ensuring that each damage category has a balanced proportion of data and contributes equally to model training. Thus, applying ADASYN not only improves the model's generalization but also enhances the stability of classification results under imbalanced data.

.TABEL IV
DECISION TREE AFTER ADASYN

Damage	Accuracy	Precission	F1-Score	Recall
Light	73.97	83.58	84.14	84.70
Moderate	73.97	68.63	68.70	68.77
Severe	73.97	69.30	68.74	68.20

TABEL V
RANDOM FOREST AFTER ADASYN

Damage	Accuracy	Precission	F1-Score	Recall
Light	81.79	89.54	91.18	92.88
Moderate	81.79	75.49	76.43	77.39
Severe	81.79	79.94	77.31	74.84

TABEL VI
GRADIEN BOOSTING AFTER ADASYN

Damage	Accuracy	Precission	F1-Score	Recall
Light	72.91	82.77	81.93	81.10
Moderate	72.91	66.47	70.46	74.96
Severe	72.91	70.03	66.01	62.43

Based on the test results, the Random Forest algorithm performed best among the three algorithms. This model achieved an accuracy of 81.79%, with high Precision, F1-score, and Recall values, especially for the light damage category, namely 89.79%, 91.18%, and 92.88%, respectively. In fact, for moderate and heavy damage, this model still maintained consistent performance, with F1 scores of 76.43% and 77.31%, respectively. These results indicate that Random Forest not only recognizes patterns in the majority of damage well, but is also quite effective at identifying minority damage after data balancing. On the other hand, the Gradient Boosting algorithm achieved an accuracy of 72.91%, the lowest of the three algorithms. Although this model still performed exceptionally well for light damage (F1-score of 81.93%), performance declined for moderate and severe damage, with F1-scores of 70.46% and 66.01%, respectively. The recall value for severe damage was only 62.43%, indicating that this model is less sensitive to it, which is important information for disaster mitigation.

Based on the evaluation results, Random Forest is the most effective algorithm in this study. With high accuracy and consistent performance across all damage levels, Random Forest produces more accurate and reliable classifications on data balanced using the ADASYN technique. Therefore, this algorithm is recommended as the primary model for developing a system to predict the level of building damage from earthquakes.

The culmination of this research was the Stacking Ensemble model, which combined Decision Tree, Random

Forest, and Gradient Boosting, successfully achieving 82.54% performance. This indicates improved performance of the individual models. Logistic Regression, as a meta-learner, plays a crucial role in determining the contribution weights of each base model. Through this training process, the meta-model can learn from the weaknesses and strengths of each base model, resulting in more accurate and stable final predictions. The following table shows a Stacking Ensemble model that does not employ ADASYN data balancing:

TABEL VII
STACKING RESULTS WITHOUT ADASYN

Damage	Accuracy	Precission	F1-Score	Recall
Light	72.44	66.78	54.95	92.88
Moderate	72.44	72.74	77.67	77.39
Severe	72.44	73.10	66.74	61.74

In this study, the hyperparameter tuning process was performed using the Grid Search Cross-Validation (GridSearchCV) method available in the Scikit-learn library. This approach works by testing all predetermined hyperparameter combinations and assessing model performance using cross-validation. Through this process, the parameter combination that yields the best performance on the training data is obtained [19].

Evaluation during the tuning process uses a 3-fold cross-validation scheme, where the dataset is divided into three equal parts. Each part is alternately used as validation data, while the other two parts are used for training. The average result of the three iterations is then used as a reference to assess the overall model performance. The primary metric used in the evaluation is the weighted F1-score (f1_weighted), as it provides a balanced assessment of precision and recall on imbalanced datasets [20].

This approach was chosen because it provides a systematic and comprehensive evaluation of each parameter combination, thereby reducing the risk of overfitting and increasing the reliability of the prediction results. Thus, the hyperparameter tuning process not only optimizes model performance but also ensures the stability and generalizability of the earthquake-induced building damage classification system developed in this study.

TABEL VIII
STACKING RESULTS USING TUNING

Damage	Accuracy	Precission	F1-Score	Recall
Light	83.01	91.77	91.63	91.50
Moderate	83.01	75.77	77.73	79.80
Severe	83.01	81.78	79.01	77.75

Overall, these tuning results demonstrate that selecting the correct parameter values significantly improves model performance. The use of Grid Search Cross-Validation proved effective in finding the best configuration that balances bias and variance, and improves classification performance on imbalanced datasets. The models with the best parameters from each algorithm were then used as base learners to form a stacking ensemble, with predictions from the three base models (Decision Tree, Random Forest, and Gradient Boosting) combined using Logistic Regression as the meta-learner. This approach aims to leverage the strengths of each algorithm in a complementary manner, resulting in a final model that is more accurate, stable, and better able to generalize to earthquake-damaged building data.

Feature importance is a model interpretability approach that measures the relative contribution of each feature to a machine learning model's predicted outcome. In other words, feature importance describes the extent to which an input variable contributes to shaping the model's decisions or outputs. This approach is essential for highly complex models such as ensemble methods (e.g., Random Forest, Gradient Boosting, and Stacking Ensemble), where the relationship between variables and predicted outcomes cannot be directly observed [21].

In practice, feature importance is used to assess the relevance of each feature and to assist the feature selection process, which selects the most informative feature subset to make the model more straightforward, more efficient, and easier to interpret without significantly reducing predictive performance. The method for calculating feature importance can vary depending on the model type. In decision tree-based models such as Random Forests or Gradient Boosting, measurements are made using mean decrease in impurity or permutation importance, which assesses the change in prediction error when the value of a feature is randomized. Meanwhile, in agnostic models, techniques such as SHAP (Shapley Additive Explanations) provide an assessment of the marginal contribution of each feature to the final prediction, accounting for feature interactions [22].

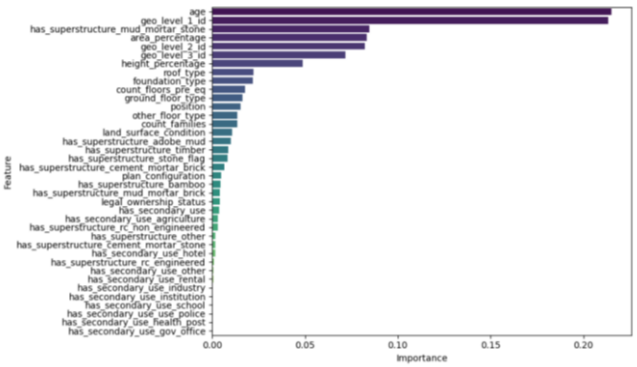

Figure 4. Combined Feature Importance

In this research, identifying the most influential features is a crucial step toward understanding the data characteristics and the model's decision-making mechanisms. Therefore, a feature importance analysis was conducted to identify the key features that significantly predict the level of earthquake-induced building damage.

The results of the feature importance analysis indicate that only a few features significantly contribute to classification. Features with importance values above the threshold of 0.03 are considered to have an important role in distinguishing the level of building damage. This threshold value was determined heuristically to retain features with significant contributions to model performance, as similar approaches have been applied in several previous studies [23], [24]. Table 9 presents the 10 main features with the most significant influence on model performance.

TABEL IX
10 MAIN FEATURES OF FEATURE IMPORTANCE

Feature Name	Reason
Age	The age of a building has the most decisive influence on the extent of damage. Older buildings generally experience reduced structural integrity due to material degradation, making them more susceptible to severe damage during an earthquake.
Geo_level_1_id	Indicates regional locations related to the geological conditions and level of seismic activity of an area. This factor helps identify differences in risk between major earthquake zones.
Has_superstructure_mud_mortar_stone	The material used to construct the walls or the main structure plays a significant role in a building's durability. Structures made of mud or mortar are more susceptible to damage from earthquake vibrations.
Area_percentage	The relative size of a building's land area is related to its structural strength. Buildings on small plots tend to be sturdier than those on larger plots with similar characteristics.
Geo_level_2_id	Represents the subregional location level that provides local variations in earthquake impact intensity, thus helping the model recognize more specific spatial patterns.
Geo_level_3_id	It is the most detailed representation of geographic location, which is important for micro-area-based predictions because it can capture differences in risk across small areas within a single zone.
Height_percentage	The relative height of a building affects its structural stability. The taller the building, the greater the inertial forces acting on it, increasing the risk of instability during an earthquake.
Roof_type	The type of roof affects load distribution and the building's center of gravity. Heavy roofs tend to increase the lateral forces a structure experiences.
Foundation_type	Foundations determine a building's ability to withstand ground vibrations. Strong and resilient foundations can reduce the risk of damage from seismic vibrations.
Count_floors_pre_eq	The number of stories serves as an indicator of potential structural damage. Multi-story buildings are more vulnerable to shear forces and bending moments during an earthquake.

From these results, it can be concluded that the building age (Age) is the most important variable, followed by geographical factors (Geo_level_1_id, Geo_level_2_id, and Geo_level_3_id) and structural characteristics such as the primary material (Has_superstructure_mud_mortar_stone) and building dimensions (Height_percentage, Area_percentage). This shows that the level of building damage is influenced not only by physical and material factors but also by environmental and geological conditions in the area where the building is located.

IV. CONCLUSION

This study shows that applying the Ensemble Stacking method, which combines the Decision Tree, Random Forest, and Gradient Boosting algorithms, significantly improves the performance of earthquake building damage classification. Based on the evaluation results, the Random Forest model

showed the most stable performance before data balancing, with an accuracy of 72.07%. After applying the ADASYN technique, the performance of all models consistently improved, especially for minority classes such as light damage. This improvement reached an optimal point in the stacking model after balancing, with an accuracy of 83.01%, indicating that the combination of the three algorithms with Logistic Regression as a meta-learner provides more accurate predictions and better generalization. These findings confirm that the ensemble stacking approach has great potential in supporting the development of data-driven building damage prediction systems.

In line with these results, the model also has the potential to be integrated into a Disaster Information System to support faster, more accurate post-earthquake decision-making. Predictions from the stacking model can be presented in an interactive dashboard, a GIS-based risk map, or an automated prediction system that displays building damage levels in real time. Thus, this model not only offers increased classification accuracy but also provides important applied value for data-driven emergency response and disaster mitigation systems [25].

The performance improvements achieved in this study are inseparable from the combination of technical strategies implemented, ranging from improving data quality to algorithm optimization. Data balancing techniques such as ADASYN help improve class distribution by generating synthetic samples from minority classes, enabling patterns to be learned more representatively [26]. Other preprocessing steps, such as normalization, feature selection, and data transformation, also reduce noise and improve training stability. From an algorithmic perspective, applying ensemble methods such as bagging, boosting, and especially stacking allows combining the strengths of several base models to improve generalization. Hyperparameter tuning in algorithms such as Random Forest, Gradient Boosting, and meta-learners then improves model performance by producing optimal parameter configurations [27].

Furthermore, applying data balancing and hyperparameter tuning significantly improved the model's final performance. ADASYN balances the class distribution so the model can better learn minority patterns, resulting in improved accuracy, recall, and F1-score. Hyperparameter tuning, on the other hand, ensures that critical parameters such as n_estimators, max_depth, and learning_rate are optimally configured to achieve a balance between bias and variance. The combination of these two processes results in a model that is more stable, accurate, and effective at learning data patterns than before balancing and tuning.

However, it is important to note that stacking models still has the potential for overfitting, especially when the meta-learner used is too complex. A complex meta-learner can capture noise in the base model's predictions and overfit the training data, thereby degrading performance on new data. To mitigate this risk, simpler meta-learner models, such as Logistic Regression, are generally preferred for their ability to maintain stable generalization.

Logistic Regression's suitability as a meta-learner is further strengthened when compared to more complex models such as XGBoost or Neural Networks. Logistic Regression has low complexity and minimal risk of overfitting, allowing it to combine base model predictions stably. Conversely, complex meta-learner models tend to be more flexible but are susceptible to overfitting to noise in meta-level data, especially when data is limited. Therefore, selecting Logistic Regression in this study is the right decision to balance model accuracy and generalization.

BIBLIOGRAPHY

[1] Badan Geologi. (2021). Peta sumber dan bahaya gempa Indonesia 2021. Pusat Vulkanologi dan Mitigasi Bencana Geologi, Kementerian Energi dan Sumber Daya Mineral. https://www.esdm.go.id

[2] Badan Meteorologi, Klimatologi, dan Geofisika. (2022). *Informasi gempa terkini dan sesar aktif di Indonesia*. https://www.bmkg.go.id.

[3] Winarsih, S., et al. (2025). Optimizing earthquake damage prediction using particle swarm optimization-based feature selection. *Jurnal Informatika dan Komputer, 11*(1), 77–86.

[4] Dachi, M. A., & Sitompul, O. S. (2023). Penerapan metode ensemble learning untuk klasifikasi data menggunakan stacking, bagging, dan boosting. *Jurnal Teknologi Informasi dan Ilmu Komputer, 10*(2), 121–130.

[5] Joses, Y. S., Yulvida, E., & Rochimah, S. (2024). Ensemble learning menggunakan stacking untuk meningkatkan kinerja prediksi pada data tidak seimbang. *Jurnal Teknologi dan Sistem Komputer, 12*(1), 89–97.

[6] DrivenData. (2020). *Richter's predictor: Modelling earthquake damage*. https://www.drivendata.org/competitions/57/nepal-earthquake/

[7] Buhl, N. (2023). Mastering data cleaning & data preprocessing. *Encord*. Retrieved May 26, 2025, from https://encord.com/blog/data-cleaning-data-preprocessing/

[8] *Dibimbing.id. (2025, Maret 24). One Hot Encoding adalah: Arti, Manfaat, dan Penerapannya. Retrieved May 26, 2025, dari https://dibimbing.id/blog/detail/one-hot-encoding-adalah*

[9] Monika, A. P., Risti, F. E. P., Binanto, I., & Sianipar, N. F. (2023). Perbandingan algoritma klasifikasi Random Forest, Gaussian Naive Bayes, dan K-Nearest Neighbor untuk data tidak seimbang dan data yang diseimbangkan dengan metode Adaptive Synthetic pada dataset LCMS tanaman keladi tikus. *Jurnal Seminar Nasional Teknik Elektro, Informatika & Sistem Informasi (SINTaKS)*, 3–7.

[10] M. Ibrahim, "Evolution of Random Forest from Decision Tree and Bagging: A Bias–Variance Perspective," *Dhaka University Journal of Applied Science and Engineering*, vol. 7, no. 1, pp. 66–71, 2022. doi: 10.3329/dujase.v7i1.62888

[11] R. Zuhri, Kusrini, and D. Ariatmanto, "Analisis perbandingan algoritma klasifikasi untuk identifikasi diabetes dengan menggunakan metode Random Forest dan Naive Bayes," *Jurnal Inovasi Teknologi dan Sains (JINTEKS),* vol. 4, no. 2, pp. 222-230,2022. [Online]. Available: https://www.jurnal.uts.ac.id/index.php/JINTEKS/article/view/5146

[12] L. W. Rizkallah, "Enhancing the performance of gradient boosting trees on regression problems," *Journal of Big Data*, vol. 12, art. no. 35, pp. 1–14, 2025. doi: 10.1186/s40537-025-01071-3

[13] W. N. Ismail and H. A. Alsalamah, "GA-Stacking: A New Stacking-Based Ensemble Learning Method to Forecast the COVID-19 Outbreak," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 202–210, 2023.

[14] Swaminathan, S., & Tantri, B. R. (2024). Confusion matrix-based performance evaluation metrics. *African Journal of Biomedical Research, 27*(4S), 4023–4031.

[15] Abubakar, P. (2025, May 8). *Evaluation metrics in machine learning: Accuracy, precision, recall & f1-score*. Medium.

https://medium.com/@abubakarp789/evaluatiom-metrics-in-machine-learning-accuracy-precision-recall-f1-score-c4c4e553677a

[16] Haya, A., & Ramme, M. Y. (2024). Penerapan algoritma stacking ensemble machine learning berbasis pohon untuk prediksi penyakit diabetes. *Prosiding Seminar Nasional Sains Data, 4*(1), 954–961.

[17] Ghimire, S., Gueguen, P., Giffard-Roisin, S., & Schorlemmer, D. (2022). Testing machine learning models for seismic damage prediction at a regional scale using a building damage dataset collected after the 2015 Gorkha, Nepal earthquake. *Earthquake Spectra, 38*(4), 2970–2993.

[18] M. Ahmed, A. Khan, and S. Hussain, "An improved adaptive synthetic sampling approach for imbalanced data classification," *Expert Systems with Applications*, vol. 206, p. 117816, 2022, doi: 10.1016/j.eswa.2022.117816

[19] E. Elgeldawi and A. M. Zaki, "Hyperparameter Tuning for Machine Learning Algorithms: A Comprehensive Comparative Analysis," *Informatics*, vol. 8, no. 4, p. 79, 2021. [Online]. Available: https://doi.org/10.3390/informatics8040079

[20] A. Ben-David, D. Lustgarten, and Y. Koren, "High Per Parameter: A Large-Scale Study of Hyperparameter Tuning for Machine Learning Algorithms," *Algorithms*, vol. 15, no. 9, p. 315, 2022. [Online]. Available: https://www.mdpi.com/1999-4893/15/9/315

[21] M. Saarela and S. Jauhiainen, "Comparison of feature importance measures as explanations for classification models," *SN Applied Sciences*, vol. 3, no. 2, pp. 41–48, 2021, https://doi.org/10.1007/s42452-021-04148-9.

[22] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, vol. 10, no. 5, p. 593, 2021, https://doi.org/10.3390/electronics10050593

[23] M. I. Prasetiyowati, N. U. Maulidevi, and K. Surendro, "Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest," *Journal of Big Data*, vol. 8, no. 1, p. 84, 2021.

[24] A. Alsahaf, A. A. Bakar, and Z. A. Othman, "A framework for feature selection through boosting," *Expert Syst. Appl.*, vol. 189, p. 116140, 2022.

[25] M. I. Prasetiyowati, N. U. Maulidevi, and K. Surendro, "Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest," *Journal of Big Data*, vol. 8, no. 84, pp. 1–24, 2021.

[26] C. Arnold, "The role of hyperparameters in machine learning models and how to tune them," *Political Science Research and Methods*, vol. 12, no. 4, pp. 841–848, 2024, doi: 10.1017/psrm.2023.61.

[27] D. V. Ramadhanti, "Perbandingan SMOTE dan ADASYN pada data imbalance," *Jurnal Gaussian*, vol. 11, no. 4, pp. 503–510, 2022.