CLASSIFYING EARTHQUAKE DAMAGE IN NEPAL:

A COMPARITIVE STUDY OF TREE-BASED ALGORITHMS AND DEEP LEARNING

DANIEL NOUMON

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2076640

COMMITTEE

Dr. Mojtaba Rostami Kandroodi
Prof. Dr. Louwerse

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

June 23th, 2023

WORD COUNT

8662

ACKNOWLEDGMENTS

Most importantly, I want to thank my family and girlfriend for their relentless support during the thesis writing process and my academic journey in general. Additionally, I want to thank Dr. Mojtaba Rostami Kandroodi for his clear guidance and devoted supervision. I thoroughly enjoyed immersing myself in the study of artificial intelligence, exploring both a wide range of theoretical background as well as its practical implications and applications.

CLASSIFYING EARTHQUAKE DAMAGE IN NEPAL:

A COMPARITIVE STUDY OF TREE-BASED ALGORITHMS AND DEEP LEARNING

DANIEL NOUMON

Abstract

The 2023 earthquake in Turkey and Syria shook the world and reminded society of its relentless destruction. In exploring methods that can assist in damage assessment, management, and mitigation efforts, the utilization of artificial intelligence proves to have the potential to obtain superior results in damage classification, over the traditionally used seismic fragility functions. Classical tree-based machine learning methods have proven their efficacy, while the application of deep learning to tabular earthquake damage data has been under-explored. Specifically, specialized deep learning framework TabNet and the multi-layer perceptron are employed as these displayed potential to surpass tree-based machine learning methods. Benchmark model LightGBM portrayed the best performance among its tree-based alternatives with micro-f1 score from a cross-validation set of 77.38% and test set of 76.71%. As for the deep learning methods, TabNet achieved the highest micro-f1 scores, settling at 75.62% cross-validation and 74.84% testing results. Between the different pre-processed configurations of baseline, feature selection, and balancing, the baselines achieved the highest performance metrics across all models. Overall, LightGBM has proven superiority in classifying building damage by the earthquake in Nepal implying tree-based methods remain state-of-the-art in this domain. A wider range of specialized deep learning models, pre-processing techniques, and regularization techniques could aid in narrowing this performance gap between deep learning and tree-based methods within this domain.

DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

Data Source: The Nepal 2015 earthquake damage dataset has been directly acquired from the DrivenData platform as the data is public, free and readily available for use through their online webpage (DrivenData, 2023). The website informs that the data originates from the Kathmandu Living Labs and the Central Bureau of Statistics, institutes that operate under the

National Planning Commission Secretariat of Nepal, and was collected to stimulate public research.

There was no need for anonymization, as work on this thesis did not involve collecting sensitive data or data from human participants or animals. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. However, the institution was informed about the use of this data for this thesis and potential research publications. All figures and images used in the thesis are original and were created for this study.

Libraries and frameworks used are listed in section 4.2. Furthermore, Open Assistant, a Large Language Model, assisted in cleaning, modulating, and debugging code. In terms of writing, assistance sofware was utilized for the language of the paper. An online writing assistant tool (Grammarly spell checker) was used to improve the author's original content in terms of paraphrasing, spell checking, and grammar. No other typesetting tools or services were used.

1 INTRODUCTION

1.1 *Problem statement*

The recent catastrophe in Turkey and Syria showcased the terrifying consequences of what a natural disaster can do. Earthquakes are a common and devastating force of nature that cause considerable damage to both human lives and infrastructure. After such an event, it is crucial to evaluate the damage extent on buildings to enable appropriate response, assistance, and rehabilitation efforts after the disaster (Sharma et al., 2016). Such data is indispensable for emergency responders, property owners, facility users, and both local and state governing bodies to make knowledgeable and timely choices, according to (Mangalathu et al., 2020). The same research states that the process of visually recognizing and categorizing specific building damage demands a considerable amount of time and personnel resources, lasting for several months following such an incident. As an illustration, conducting the initial round of building damage field assessments following the 1994 Northridge earthquake took over two months (Trifunac & Todorovska, 1997). Given these rationales, it would be advantageous to have a system that can anticipate the most probable degree of damage and repair intervention required. At present, such systematic forecasts rely on seismic fragility functions (Martins & Silva, 2021).

Fragility functions rely on pre-defined mathematical models that assume specific characteristics and may not consider all factors that can impact structure vulnerability (Mangalathu et al., 2020). The limited scope

of features inherent in fragility functions creates a deficiency that prevents them from comprehensively capturing the intricate nature of real-world phenomena (Hwang et al., 2021). Conversely, A.I.-based models provide the opportunity to incorporate a wide range of collected building features (Mangalathu et al., 2020; Nateghi et al., 2011). Through this diversity patterns can be identified that traditional seismic models may not capture, leading to more accurate and nuanced risk assessments (Sajan et al., 2023).

Such a comprehensive data set was created after the earthquake that took place in Nepal in 2015, damaging more than a million buildings in the process and affecting even more lives (Lizundia et al., 2017). The research of Lizundia et al. (2017) also highlights that Nepal is located in a geographical area prone to earthquakes, with this particular earthquake constituting a magnitude of 7.8 on the Richter scale. As a result, Nepalese authorities gathered data to assist in managing highly probable future earthquake scenarios (Sharma et al., 2016). The data collected regards a wide range of building features and their damaged status which are classified in low, medium, and high degrees of impact severity (DrivenData, 2023).

When it comes to selecting an A.I.-based approach, machine learning models have emerged as a viable option, demonstrating promising results in predicting earthquake damage (Sajan et al., 2023). Several studies employed machine learning techniques for damage identification and fragility analysis and gained higher accuracy estimates than the predefined fragility functions they were compared to (Feng et al., 2020; Kiani et al., 2019; Siam et al., 2019). So far, in the previously mentioned studies, the predominantly utilized machine learning models in this area are the tree-based methods of Decision Trees, Random Forests, Light-Gradient Boosting (LightGBM), Extreme Gradient Boosting (XGBoost), and Category boosting (CatBoost). The application of deep learning techniques however, is currently absent. This is consistent with the research of Ye and Wang (2023), who state that the application of deep learning methods on tabular data is still under-explored.

Recently, several specialized deep learning architectures have been created to enhance the performance of deep learning efforts on tabular data (Borisov et al., 2022; Grinsztajn et al., 2022; Kadra et al., 2021). From these specialized frameworks, TabNet was one of the first widespread adaptations in this type of application (Borisov et al., 2022). It has an architecture proposing a sequential attention mechanism in picking a subset of meaningful semantic attributes at each decision step (Arik & Pfister, 2021). On the other hand, findings from Kadra et al. (2021) find that deep learning in the form of well-tuned multi-layer perceptrons with 9 hidden layer and 512 nodes per layer surpass both specialized deep learning frameworks and tree-based methods such as XGBoost on tabular data. This highlights

the potential for a multi-layer perceptron framework to refine the veracity of machine learning and deep learning models to structured earthquake data. Moreover, the research conducted by Grinsztajn et al. (2022) showed that both specialized deep learning and multi-layer perceptron approaches can substantially benefit from pre-processing the data.

Therefore, this thesis aims to investigate the predictive power of both the TabNet and the multi-layer perceptron deep learning frameworks, their dependence on pre-processing, and compare them to conventional tree-based machine learning models in classifying earthquake damage. Results are evaluated which can have substantial practical implications for earthquake damage assessment, management, and mitigation efforts.

1.2 *Main Research Question*

> *Does deep learning exhibit superior predictive power over machine learning tree-based models when classifying earthquake damage degrees to buildings in Nepal?*

The main research question aims to contribute an analysis of which method proves to be state-of-the-art in predicting tabular earthquake data. Parsimonious tree-based methods are compared with unconventional TabNet and multi-layer perceptrons to derive an outcome.

1.3 *Sub-Questions*

RQ1 *Which tree-based method offers the best performance?* This question aims to identify the benchmark results of conventional methods. This provides a basis for comparison with the multi-layer perceptron method.

RQ2 *Which deep learning framework proves to offer the best performance?* With this question the assessment of TabNet, a deep learning framework specialized for tabular data, versus the multi-layer perceptron framework is evaluated.

RQ3 *To what extent can pre-processing techniques improve the results of the deep learning techniques on the data?* As deep learning applied to tabular is unconventional, various pre-processing steps should be tested to find out its impact on performance. This could help in aiding the deep learning methods to surpass tree-based results.

2 LITERATURE REVIEW

2.1 *Tree-based Frameworks*

In recent years, there has been a considerable amount of research focused on analyzing and processing tabular data (Borisov et al., 2022; Gorishniy et al., 2021; Grinsztajn et al., 2022; Ye & Wang, 2023). Among these, tree-based methods have proven to be state-of-the-art. The study by Borisov et al. (2022) showed that LightGBM, XGBoost, and Catboost outperformed methods such as linear regression, logistic regression, and K-nearest Neighbors on all of the studied datasets in both accuracy and ROC-AUC tasks.

When it comes to tabular earthquake data specifically, the approach of Mangalathu et al. (2020) to classifying earthquake damage to buildings entailed comparing Decision Trees, Random Forests, and other supervised methods such as Logistic Regression and the unsupervised method of k-Nearest Neighbors. Likewise to the previously stated research, the tree-based methods applied to tabular earthquake data specifically proved to be the most effective, further supporting their viability and superiority in this particular context. This observation is corroborated by additional research conducted in this field by Sajan et al. (2023), where XGBoost emerged as the most effective tree-based algorithm for classifying building damage in earthquakes. Similair to Mangalathu et al. (2020) Decision Trees and Random Forrest were candidate methods in this study, though these demonstrated a lower level of effectiveness in the multi-class classification prediction.

2.2 *Deep Learning Frameworks*

Based on recent research, it appears that in analyzing tabular data on medium-sized datasets with 10,000 training examples, tree-based models such as Random Forests and XGBoost outperform deep learning methods (Grinsztajn et al., 2022). However, as the dataset size increases, the gap between these two types of models becomes narrower, with deep learning methods catching up to tree-based models (Grinsztajn et al., 2022). This makes deep learning especially promising for this research since the number of available instances in the Nepal earthquake dataset is 260,601 in total. According to the findings from Shwartz-Ziv and Armon (2021), which compared various specialized deep learning frameworks on tabular data, TabNet emerged as a viable option when compared to its selected alternate candidates of NODE, DNF-Net, and 1D-CNN. This finding aligns with the research conducted by Borisov et al. (2022), which supports the notion that TabNet has the capability to outperform specialized frameworks such

as VIME, DeepFM, DeepGBM, and others on tabular data. Their study provides additional validation, reaffirming the effectiveness of TabNet as a superior choice for analyzing tabular datasets.

On the contrary to the aforementioned studies, Kadra et al. (2021) found that with adequate regularization, multi-layer perceptrons are able to outperform specialized Deep Neural Network architectures on tabular data. Moreover, he found that with the optimal set of hyperparameters, even Gradient-Boosted Trees can be surpassed. In this study, a proposed optimal configuration of the multi-layer perceptron with fixed layers and node configuration by Yoon et al. (2021) is utilized. Their approach suggests a multi-layer perceptron network with an architecture configuration of 9 feed-forward layers containing 512 units per hidden layer. Subsequently, an optimization hyperparameter search space is traversed to fine-tune the model.

Alternatively, instead of initializing a fixed network size, Kadra et al. (2021) proposed that other network sizes should be traversed by automated search for further enhancement. The application of joint architecture and hyperparameter optimization should be explored, since the optimal size of the network can vastly differ per problem (Apolloni & Ronchini, 1994). As effective manual refinement of architectures requires extensive expertise and a substantial amount of time, automated search provides to be an efficient alternative. A Bayesian optimization strategy can be an efficient yet effective solution for finding the optimal network architecture (Snoek et al., 2012). In addition, Bayesian optimization is deemed superior to conventional Gridsearch and Randomizedsearch in hyperparameter tuning for machine learning according to the research by (Turner et al., 2021). This method depends on metrics obtained from previously trained models to steer the decision of which architectures to establish and train in successive steps (Yang & Shami, 2020). The utilization of prior information typically helps diminish the number of architectures to educate in future iterations.

2.3 *Pre-processing techniques*

Besides the configuration of the multi-layer perceptron size and hyperparameters, pre-processing the data and the selection of features is pivotal on the outcome of deep learning models (Grinsztajn et al., 2022). The research states that uninformative features can have a consequential impact on the performance of these models. They have also shown that deep learning methods are more vulnerable to the negative effects of uninformative features than tree-based methods. Therefore, when working with tabular data, it's important to carefully consider the role of uninformative features and choose the appropriate model for the dataset size and feature quality.

Another impactful component of pre-processing tabular data for deep learning is data balancing (Johnson & Khoshgoftaar, 2019). According to their study, when dealing with imbalanced data, over- and undersampling techniques can prevent models from disproportionate bias towards the majority class, which can be detrimental for model performance.

2.4 *Summary*

In short, tree-based algorithms performances are state-of-the-art in classifying earthquake data. decision trees, random forests, XGBoost, LightGBM and Catboost all prove to be viable candidates to set the state-of-the-art baseline in the tabular data domain. Nevertheless, large datasets of 50,000+ instances make deep learning techniques competitive. Specialized deep learning frameworks as well as optimized multi-layer perceptron models have under-explored potential to outperform tree-based methods on tabular data. To unlock that potential, feature selection and balancing the data have proven to be pivotal strategies.

3 METHODOLOGY

3.1 *Dataset*

After the 2015 Earthquake, Kathmandu Living Labs and the Central Bureau of Statistics, institutes that operate under the National Planning Commission Secretariat of Nepal, conducted surveys aiming to gather rigorous and valuable information on earthquake impacts (Ghimire et al., 2022). The data is freely distributed by DrivenData (2023), a platform for open-source projects in data science. In total, the full dataset contains 260,601 instances that get described by 38 features, as well as the unique identification number of a building. The original distribution of the labels within the full dataset has a representation of 25,124 low damage grades (grade 1, representing 10%), 148,259 medium damage grades (grade 2, representing 57%), and 87,218 high damage grades (grade 3, representing 33%).

It should be noted that the website does not offer any additional details per damage grade beyond these classifications. By categorizing the damage into multiple levels, it allows for a more specific analysis of the varying degrees of impact on different structures and areas. As previously stated in the problem statement, understanding the specific distribution of damage across these grade levels enables researchers and policymakers to prioritize resources and interventions based on the severity and urgency of the affected areas. While a binary classification of no damage versus damage may provide a simplified view, the multi-class approach captures

the complexity of the situation and provides more valuable insights for mitigation and recovery efforts.

Even though previously mentioned studies highlight deep learning excellation on large datasets, the full extent of the dataset size is not utilized in this study. Due to computational constraints and time limitations, the dataset size was reduced to 50,001 instances with the same distribution to enable feasible yet representative processing and analysis of the data. The first grade is represented by 4808 instances, grade 2 is represented by 28,366 instances, and the third is represented by 16,827 instances respectively. There are no missing values in the entire dataset. 9 features are numeric, 9 are categorical, and 22 are boolean. The features include geographic region, floor count, area building footprint, building height, land surface, foundation type, roof type, ground floor type, other floor type, position, superstructure type, family count, and secondary usage. For a full list of features, see Appendix A.

3.2 *Software*

The algorithms applied in the thesis will be deployed in Python 3.8.11. Visual Studio Code (VScode) is employed as a source code editor. Packages used: `Numpy`, `Pandas`, `Matplotlib`, `Seaborn`, `Scikit-learn` (`Sklearn`), `Scikit-optimize` (`Skopt`), `Category_encoders`, `XGBoost`, `LightGBM`, `CatBoost`, `PyTorch_tabnet`, `TensorFlow`, and `Keras`.

3.3 *Setup and evaluation*

Misclassifying the damage grade of a building could have costly consequences for human health (Mangalathu et al., 2020). Emergency responders have to accurately decide where to allocate their finite resources. Therefore, in the context of this multi-class classification analysis, the evaluation metric that is picked to compare the efficacy of each model will be the micro-f1 score. The micro-f1 score summarizes the model's accuracy across all classes, whilst assigning higher weights to smaller classes to take class imbalance into account (Lipton et al., 2014). This is especially relevant to the model applications of pre-processed datasets that don't utilize data balancing. In addition, the Receiver Operating Characteristic Area Under the Curve (ROC AUC) is employed to help understand the model's ability to discriminate between the different damage grades. This method extends beyond the micro-f1 score, allowing for a more comprehensive understanding of the classifier's ability to address the challenges posed by imbalanced class distributions (Huang & Ling, 2005). A confusion matrix and classification report are employed to aid in evaluating these results.

Furthermore, the data will be split in training (80%) and test sets (20%), and a K-Fold cross-validation of 5 folds is applied to prevent overfitting whilst trying to find the best hyperparameters. This improves generalization beyond the training data by repeatedly partitioning the available data into training and validation subsets, evaluating the model's performance across multiple iterations (King et al., 2021). Both splitting and cross-validation are applied in combination with stratification, which entails that the class distribution for each fold is the same as the original distribution to warrant a fair comparison between models and ensure reproducibility. Random seeds of 42 for data utilization is implemented to aid that same purpose.

Lastly, in assessing the statistical significance of the difference between the results, approximate randomization is applied. Although the two-sample t-test is a commonly utilized method to compare machine learning performances, its assumptions assumes normality and requires homogeneity of variances (Raschka, 2020). Both of these are violated by the Bayesian search approach. Instead, approximate randomization was selected due to its ability to handle the inherent uncertainty associated with the Bayesian search framework through its robustness to violations of distributional assumptions, as stated by (Still & White, 1981). This non-parametric test does not rely on specific distributional assumptions and allows for rigorous comparisons by generating 10,000 random permutations of the data, as 10,000 permutations is deemed robust (Hayes, 1998). Hayes (1998) states that by simulating the null distribution of the test statistic under the assumption of no difference between the models, approximate randomization provides a rigorous assessment of the statistical significance of differences observed in the results. Accordingly, the resulting p-values inform of the presence of statistical significant differences between the results acquired through Bayesian search for the different candidate models.

3.4 *Algorithms*

3.4.1 *Tree-based Frameworks*

According to research of (Borisov et al., 2022; Ke et al., 2017; Mangalathu et al., 2020; Prokhorenkova et al., 2018; Sajan et al., 2023) tree-based methods of Decision Trees, Random Forests, XGBoost, LightGBM, and Catboost all demonstrate proficient capabilities in analyzing tabular datasets across different domains and within the earthquake domain specifically. As the results between these vary in the different studies, as well as differing datasets utilized, there is no clear superior method between these. Therefore, each of these models will be considered as candidates that

could set the benchmark for conventional approaches in this study. Other tree-based methods such as AdaptiveBoosting (AdaBoost) are not present in the aforementioned-studies, and therefore not taken into consideration. All tree-based methods are fine-tuned using a Bayesian optimization approach. BayesSearchCV traverses through the encompassing hyperparameter search spaces as defined by the afore-mentioned studies. These search spaces can be found in Appendix B

3.4.2 *TabNet Framework*

As described in the literature review, TabNet is selected as the specialized deep learning framework candidate in this research. Arik and Pfister (2021) state that TabNet operates by iteratively selecting important features and processing them through decision steps called "attentive transformers". They posit these decision steps are designed to capture complex dependencies and interactions within the data. It is explained that each decision step consists of a feature-wise attention mechanism followed by a feature transformer network. The attention mechanism dynamically selects relevant features based on their importance, allowing TabNet to focus on the most informative aspects of the input data.

3.4.3 *Multi-layer Perceptron Frameworks*

Specialized deep learning models, as previously stated in this research, can potentially be outperformed by well-regularized multi-layer perceptrons. Two differing strategies in forming the optimal multi-layer perceptron model will be utilized. The first multi-layer perceptron model consists of 9 hidden layers with 512 nodes per layer as proposed by Kadra et al. (2021). According to their study, a congruent hyperparameter with this architecture is AdamW, which applies decoupled weight decay. AdamW is a form of stochastic optimization that differs from the conventional Adam method by disentangling weight decay from the gradient update process (Loshchilov & Hutter, 2019). According to their research, this yields better generalization.

The following is an overview of its components:

$$\theta_{t+1,i} = \theta_{t,i} - \eta \left(\frac{1}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t + w_{t,i}\theta_{t,i} \right), \forall t$$

- $\theta_{t,i}$: The current value of the parameter θ at time step t for component i.
- η: The learning rate, which determines the step size for the parameter update.

- $\hat{v}_t$: The estimated second moment of the gradients up to time step t, with bias correction applied.
- ϵ: A small constant added for numerical stability to avoid division by zero.
- $\hat{m}t$: The estimated first moment of the gradients up to time step t, with bias correction applied.
- wt, i: The weight associated with the component i of the parameter θ at time step t.
- $\forall t$: Denotes that the parameter update is performed for all time steps t.

The AdamW optimizer updates the model's parameters using a combination of momentum and adaptive learning rates. It calculates the first and second moments of the gradients and uses these to update the parameters. The weight decay term, proportional to the learning rate, is added to the parameter update step. This helps to directly regularize the weights and mitigate the impact of large weights. By combining the benefits of adaptive learning rates from Adam and weight decay regularization, AdamW can help improve the generalization performance of the multi-layer perceptron framework and prevent overfitting (Kadra et al., 2021).

The same study on well-regularized multi-layer perceptrons utilized Cosine annealing as a learning rate scheduler in combination with AdamW. They write that using a learning rate scheduler with restarts aids in performance when keeping a fixed initial learning rate. Practices from Zimmer et al. (2021) are adopted for restarts by using an initial budget of 15 epochs and a budget multiplier of 2, as explained in the following notation.

$$\eta_t = \eta^i_{\text{min}} + \frac{1}{2}\left(\eta^i_{\text{max}} - \eta^i_{\text{min}}\right)\left(1 + \cos\left(\frac{T_{\text{cur}}}{T_i}\pi\right)\right) \tag{1}$$

- η_t represents the learning rate at iteration t.
- η^i_{min} is the minimum learning rate for the i-th cycle of the annealing process.
- η^i_{max} is the maximum learning rate for the i-th cycle of the annealing process.
- T_i represents the length of the i-th cycle.
- T_{cur} is the current iteration number within the current cycle.

The formula itself calculates the learning rate at a given iteration t within a cycle i of the annealing process. It follows a cosine-shaped curve that oscillates between η^i_{min} and η^i_{max}. As the current iteration number T_{cur} increases within the cycle, the learning rate smoothly transitions from η^i_{min} to η^i_{max} and then back to η^i_{min}. This cyclic behavior helps optimization algorithms to explore different regions of the loss landscape during training, potentially aiding convergence and preventing getting stuck in local optima (Loshchilov & Hutter, 2019). In the context of restarts, after each cycle, the scheduler restarts with a new cycle length T_i that is multiplied by 2 compared to the previous cycle, as described by (Kadra et al., 2021). This means that the length of each cycle doubles after each restart. The cosine function, $\cos(\cdot)$, inside the equation varies between -1 and 1, creating the oscillatory behavior of the learning rate. The terms involving η^i_{min} and η^i_{max} determine the range of learning rate values, and the ratio T_{cur}/T_i controls the progress within the cycle.

Finally, he states that this formula with restarts can be used in optimization algorithms to dynamically adjust the learning rate during training, creating a cyclical pattern and allowing exploration of different regions of the loss landscape. The length of each cycle doubles after each restart, providing a mechanism for gradually increasing the exploration range throughout training.

3.5 *Pre-processing Techniques*

All categorical features were target encoded to be able to run the multi-layer perceptron models. Target encoding is a method that handles high cardinality categorical variables well where there is a large number of unique categories, according to Pargent et al. (2022). Since the geospatial categories in this dataset contain 10,599 unique values, this method is more applicable than one-hot encoding. All numerical features were standardized $z = \frac{x - \mu}{\sigma}$ to improve the convergence of optimizer in the multi-layer perceptron models, as shown by Dzierżak (2019). As described in the setup and evaluation section, the training and testing data are split into 80% training and 20% testing data and 5-Fold cross-validation combined with Bayesian search is applied. As explored in the theoretical framework, the effects of applying two pre-processing methods are investigated to TabNet and the multi-layer perceptrons.

Baseline	**Customized**
No feature selection	Feature selection
No data balancing	Data balancing

By exhausting all 4 possible configurations of these techniques a robust comparison is employed of the impact of each configuration.

Baseline	FS	DB	FS+DB

Abbreviations:

- FS: Feature Selection
- DB: Data Balancing

3.5.1 *Feature Selection*

Even though wrapper methods in feature selection are known to have high certainty in finding the most informative features for a model, they are computationally expensive (Sánchez-Maroño et al., 2007). Consequently, a filter method is applied to find the correlation between the damage grade (dependent variable) and each feature per building (independent variable). According to Bommert et al. (2020), feature selection method based on Random Forests filter method permutation allowed for the best estimator.

The permutation importance is calculated by analyzing the out-of-bag (OOB) instances for each tree, which refers to the instances that were not utilized during the tree fitting process, as explained by (Altmann et al., 2010). It is further explained that for each tree's OOB instances, feature X_i is permuted and the resulting permuted instances are then classified by the corresponding trees. By comparing the resulting classification micro-f1 score with the micro-f1 obtained without permuting the feature, the score of the permutation importance filter is determined. This score is represented by the decrease in classification accuracy from the original OOB instances to the permuted instances. As indicated by Izenman (2008), features that play a crucial role in class prediction are identified by causing a significant decrease in accuracy when their relevant information is not available due to the permutation of the feature.

3.5.2 *Data Balancing*

Johnson and Khoshgoftaar (2019) found that oversampling is more effective than undersampling due to the prevention of loss of information. That being said, there are 260,601 instances in the full available dataset from the Driven Data website, with the least represented target class (Damage grade 1) represented by 22,000 instances. As this study only utilizes 80% of 50,001 instances for training the models, 40,001 instances make up the data for cross-validation. A balanced subsample of 40,001 instances can be maintained simply by utilizing 13,337 (40,001 /3 = 13,337) instances of

each target class, thereby preventing loss of information by sampling an equal distribution of the full dataset.

3.6 *schematic*

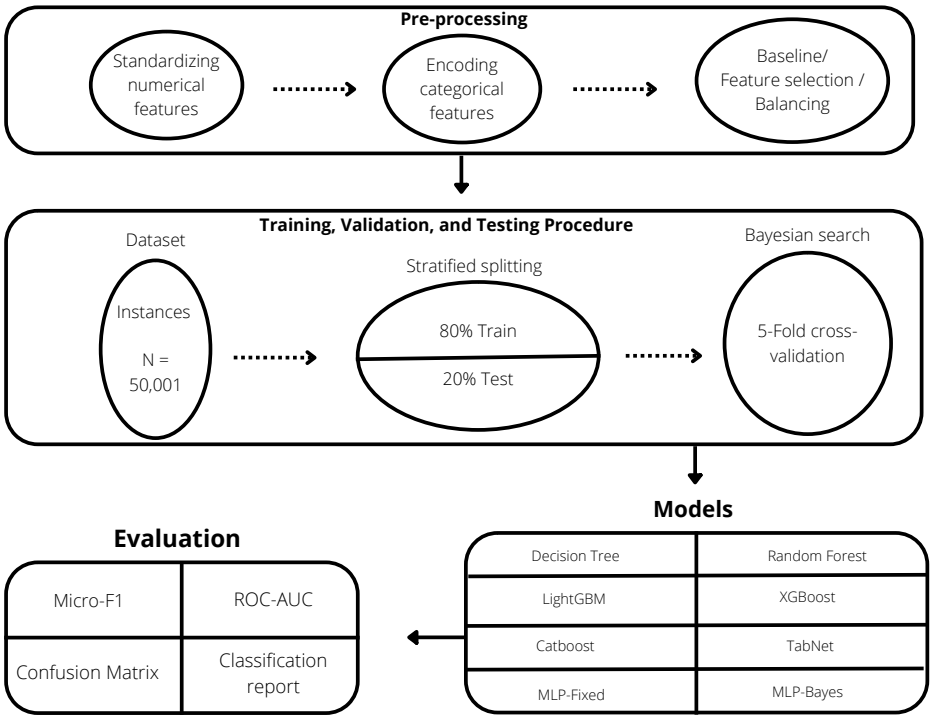

Figure 1: Methodology flowchart

4 RESULTS

4.1 *Tree-based methods*

4.1.1 *5-Fold cross-validation scores*

The highest-scoring iteration from the 200 iterations of Bayesian search is presented per model, which includes the mean score of its 5-fold cross-validation scores and the corresponding standard deviation. The LightGBM baseline model provided the best performance on the cross-validation sets, whereas the worst performing model can be ascribed to Catboost.

Table 1: Bayesian Optimized cross-validation micro-f1 Scores per Configuration

Model	Base	FS	DB	FS+DB
Decision Tree	75.72 ± 0.25	**75.83 ± 0.24**	73.80 ± 0.41	75.76 ± 0.28
Random Forest	**76.70 ± 0.40**	76.62 ± 0.40	76.53 ± 0.44	76.51 ± 0.43
LightGBM	**77.38 ± 0.22**	77.15 ± 0.21	74.76 ± 0.31	75.97 ± 0.28
XGBoost	**77.23 ± 0.21**	77.16 ± 0.18	77.05 ± 0.22	77.10 ± 0.21
CatBoost	73.87 ± 0.49	**73.88 ± 0.40**	73.13 ± 0.54	73.18 ± 0.51

4.1.2 *Test set scores*

Even though scores remain similair, small drops in results can be noted across each configuration, which implies good generalization capabilities to the unseen data on the hold-out test set.

Table 2: Test Set micro-f1 Scores per Configuration

Model	Base	FS	DB	FS+DB
Decision Tree	74.92	**74.95**	72.71	74.72
Random Forest	**76.37**	76.10	75.50	75.84
LightGBM	**76.71**	76.67	69.86	71.21
XGBoost	76.51	**76.60**	76.21	76.55
CatBoost	72.77	**72.93**	72.03	72.05

Based on these validation and test scores, it can be observed that the effectiveness of feature selection varies across models. In some cases, applying feature selection slightly improves the performance, while in others, the base configuration performs better. Data balancing on the other hand, does not lead to any enhancement over the base model.

4.1.3 *Top tree-based performance in LightGBM Baseline*

When taking a closer look at the best performing tree-based method, LightGBM, the Reicever Operating Characteristic-Area Under the Curve (ROC-AUC) score per class makes it apparent that the model is slightly better in classifying buildings with low damage grades and high damage grades. Even though still a reasonably good score, the model is not as accurate in distinguishing buildings graded as moderately damaged compared to non-moderately damaged buildings. The macro-average score is slightly lower than the micro-average score, as it is more influenced by the performance of the majority class (damage grade 2).

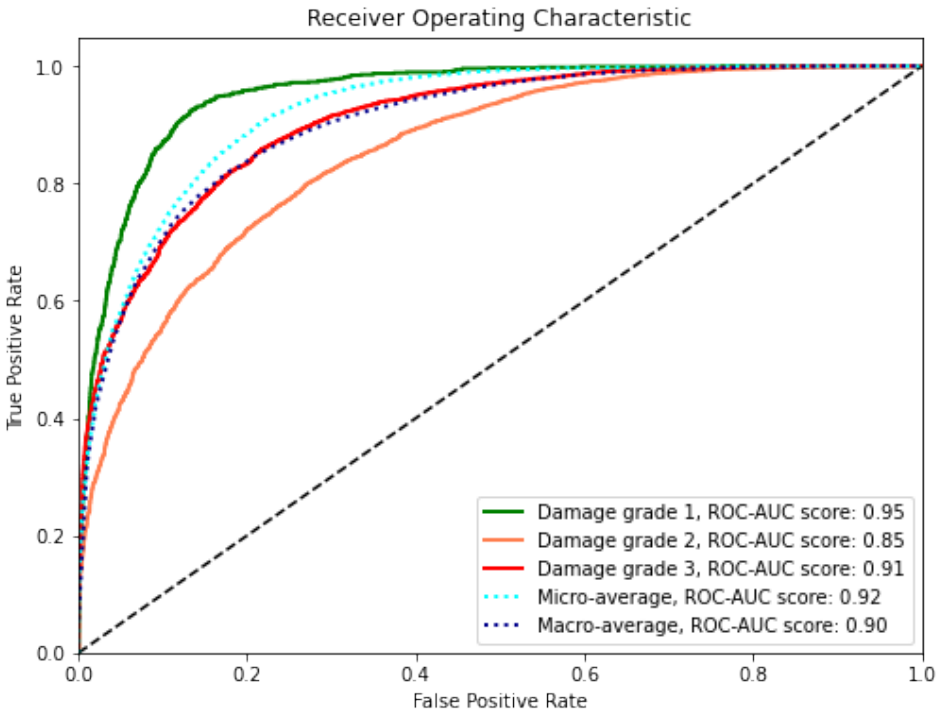

Figure 2: LightGBM ROC-AUC per class

The confusion matrix aids in explaining the varying ROC-AUC scores for classifying earthquake damage grades. For damage grade 1, the matrix indicates 559 correct predictions and a similar number of false negatives (403). Notably, Damage grade 2 showcases a high number of 4,756 accurate classifications along with a high number of false positives (1,385) and false negatives (917). Damage grade 3 exhibits moderate performance with 2,357 correct predictions, containing a moderate amount of false positives (672) and false negatives (1,007). The contrast between these scores and the

ROC-AUC can be ascribed to the difference in data distribution.

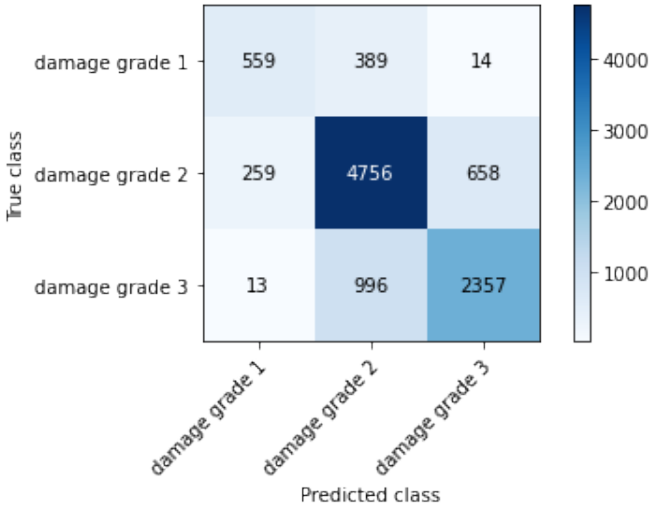

Figure 3: LightGBM Confrontation Matrix

For additional detailed evaluation metrics per class, the following classification report provides insight.

Damage Grade	**Precision**	**Recall**	**f1-score**
1	67	58	62
2	77	84	81
3	78	70	74
Accuracy		77	

Table 3: LightGBM Classification Report in %

The precision scores indicate the model's ability to correctly identify instances of each damage grade. The model performs roughly the same on medium and high damage grades, whereas the low damage grade performs worst. Interestingly, there is a clear divide in performance in all of the three classes when it comes to recall. Low damage grade exhibits a recall of 58%, indicating that the model correctly identified 58% of the actual grade 1 buildings. This detriment comes at the benefit of identifying medium damage grades which sees a substantial raise in relative performance for the recall metric, finalizing at 84%. High damage grade performs moderate at 70%. Therefore, the f1-score for classifying low damage grades

is worst at 62%, medium damage grade is best at 81%, and high damage grade performs reasonably at 74%. Since the data was imbalanced in this configuration, the overall accuracy is not an average of all three F1 scores, but indicates the model's capability to correctly classify 77% of all buildings. This is mostly positively impacted by the majority class damage grade 2.

4.2 *Deep learning methods*

4.2.1 *5-Fold cross-validation scores*

In line with the evaluation process applied to the tree-based methods, the performance of the deep learning models is assessed through the same methodology. Top performing iteration mean test scores are found alongside their corresponding standard deviations. From the different deep learning configurations, the TabNet base model achieved the highest micro-f1 score amongst its competitors. As seen below, the multi-layer perceptron configurations performed substantially worse. However, when comparing the baseline models with their pre-processed alternatives, all configurations achieve competitive scores.

Table 4: Bayesian Optimized Validation Set Micro-f1 Scores per Configuration

Model	Base	FS	DB	FS+DB
TabNet	**75.62 ± 0.42**	74.83 ± 0.34	73.39 ± 0.37	74.76 ± 0.36
MLP-Fixed	**68.08 ± 1.30**	67.76 ± 1.06	67.53 ± 1.18	67.51 ± 1.15
MLP-Bayes	**69.39 ± 0.21**	69.18 ± 0.13	68.76 ± 0.20	68.15 ± 0.13

4.2.2 *Test Set scores*

When assessing the generalization performance of TabNet, a small reduction in the performance metric on the test set can be seen. On the contrary, both multi-layer perceptron methods perform slightly better on the test set, with incremental increases in the micro-f1 scores.

Table 5: Test Set Micro-f1 Scores per Configuration

Model	Base	FS	DB	FS+DB
TabNet	**74.84**	74.15	72.02	73.76
MLP-Fixed	**68.49**	66.93	63.99	65.12
MLP-Bayes	**69.66**	69.43	68.32	68.03

When looking at the effect of the pre-processing methods applied to deep learning, a drop in performance can be seen across the entire spectrum. Especially balancing the data substantially lowers the ability to correctly classify building damages.

4.2.3 *Top deep learning-based performance in TabNet Baseline*

To assess the classification performance of TabNet, ROC-AUC scores per class are compared. The results revealed varying levels of performance across different damage grades. Notably, the model achieved the highest ROC-AUC score for damage grade 1, indicating a strong ability to accurately classify buildings with low damage. For medium damage grades, the model obtained a slightly lower ROC-AUC score, suggesting a relatively good performance but with a marginally reduced accuracy compared to low damage grades. Similarly, the model exhibited a commendable ROC-AUC score for high damage grades, indicating effective classification albeit with a lower precision compared to grade 1. Furthermore, the micro-average ROC-AUC score considering the overall performance of the model across all damage grades. It scored slightly higher than, the macro-average ROC-AUC score, which represents an average evaluation of the model's classification performance, with equal weight assigned to each damage grade. The lower macro-average score highlights potential challenges in accurately classifying the grade with the highest number of instances, grade 2.

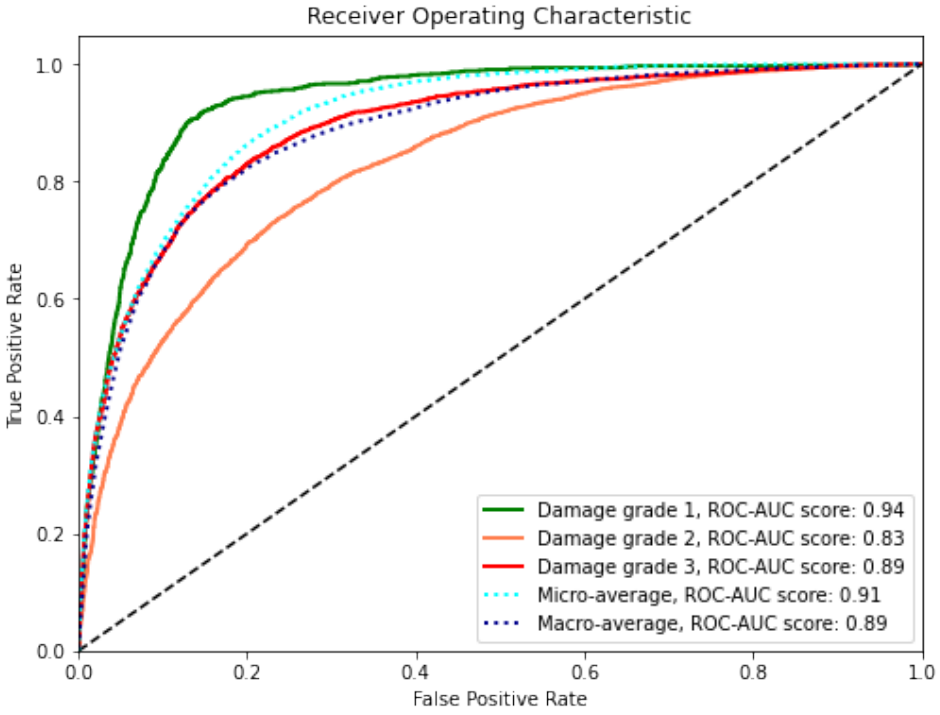

Figure 4: TabNet ROC-AUC per class

In order to gain deeper insights into the varying performance of TabNet in classifying the different damage grades, a confusion matrix is generated. For damage grade 1 the model achieved a high true positive count (403) while exhibiting false positives (259) and false negatives (559). Damage grade 2 displayed a substantial number of false positives (1,502) and false negatives (984). Moreover, damage grade 3 experienced (755) false negatives (973) alongside a relatively high true positive count (2,393).

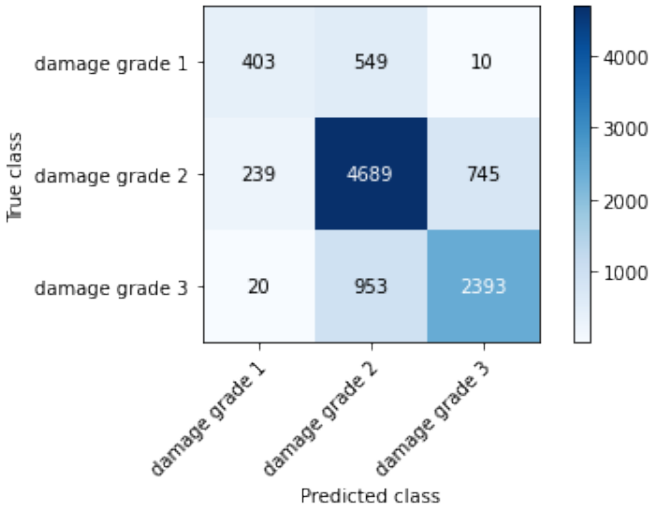

Figure 5: TabNet Confusion Matrix

Damage Grade	Precision	Recall	f1-score
1	61	42	50
2	76	83	79
3	76	71	73
Accuracy		75	

Table 6: TabNet Classification Report in %

The model demonstrates similar performance for medium and high damage grades, while it exhibits lower precision for the low damage grade. Additionally, there are notable differences in recall across all three classes. Specifically, the recall for the low damage grade is 42%, indicating that the model accurately identifies 42% of the actual grade 1 buildings. In contrast, there is a substantial improvement in relative performance for medium damage grades, with an 83% recall rate, while the high damage grade achieves a moderate recall of 71%. The f1-scores for classification are 50% for low damage grades (worst), 79% for medium damage grades (Best), and 73% for high damage grades (reasonably good). It is important to acknowledge that the overall accuracy, which is 75%, is not an average of all three F1 scores due to imbalanced data. However, it indicates the model's capability to correctly classify 75% of all buildings. This accuracy rate is influenced positively by the overrepresentation of damage grade 2 instances and negatively impacted by the underrepresentation of damage grade 1 instances.

4.3 *Comparison of best performing configuration of all models*

According to Shwartz-Ziv and Armon (2021), comparing models requires capturing details about the ease with which the appropriate hyperparameters are found. Consequently, the hyperparameter tuning of the top scoring configuration of each model is plotted below. For the tree-based methods, this implies that the feature selected Decision Tree and Catboost are displayed, whereas the remaining candidates are the baselines configurations. For the deep learning models, all baselines are shown. The visualization allows for the examination of the trends and patterns exhibited by each model over the iterations, which offers insights regarding their convergence, stability, and overall performance. The ribbons correspond to the minimum and maximum scores for each of the 5-folds within the iteration.

Notably, the best performing models show the most stable convergence, starting with a mean micro-f1 score not too dissimilar from their converged scores. XGBoost and LightGBM, and TabNet prove to be the most stable out-of-the-box models, followed closely by the convergence plots of the Random Forest and Catboost models. The models with the most volatile learning curves are the Decision Trees multi-layer perceptron with fixed architecture, and the multi-layer perceptron with the Bayesian optimized architecture.

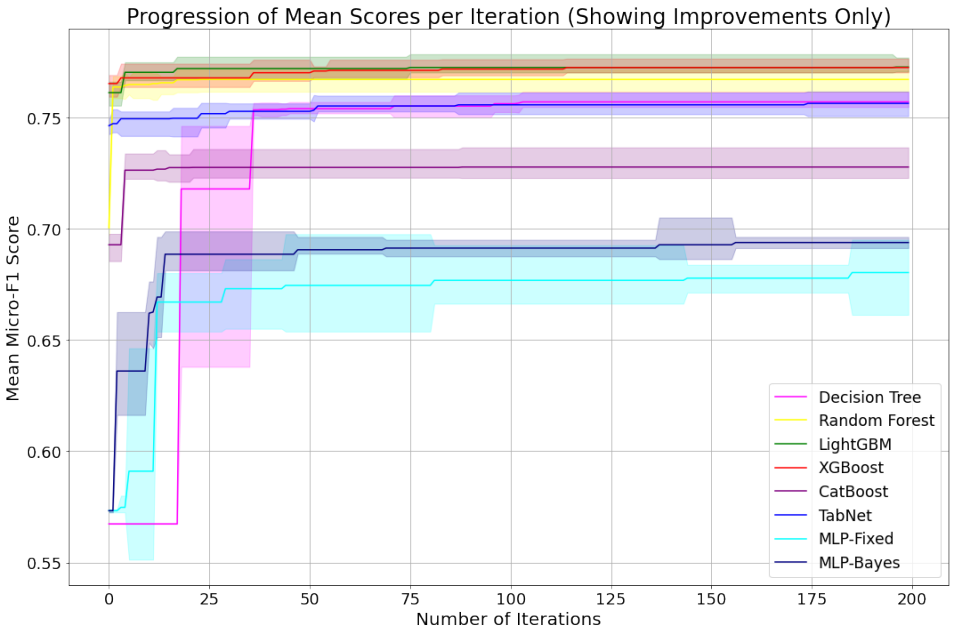

Figure 6: Model improvement comparison plot

4.4 *Statistical Comparison of Best Performing Configuration of all Models*

By applying the non-parametric approximate randomization test on the 5-Fold mean cross-validation scores, the testing for significance in difference is conducted. We define the null hypothesis (H_0) as the assumption that there is no significant difference in performance metrics between the machine learning model pairs. The alternative hypothesis (H_1) suggests that a significant difference does exist. The determination of whether to reject the null hypothesis for each model pair is based on the resulting p-value, which is influenced by the threshold significance level of 95% ($\alpha = 0.05$) for this study. Based on the resulting p-value, if the p-value is greater than alpha (α), we fail to reject the null hypothesis, indicating that observed performance differences are likely due to random chance. Conversely, if the p-value is less than alpha (α), we reject the null hypothesis, concluding that a significant difference exists in performance between the model pairs.

In the matrix, all model candidates are presented accompanied by their top performing configuration micro-f1 score in brackets. For every model pair combination, the p-value is presented. There are only three model pairs where a failure to reject the null hypothesis (H_0) can be stated. These are Decision Tree compared with TabNet, LightGBM compared with XGBoost, and MLP-Fixed compared with MLP-Bayes. This denotes that there are two best performing models, LightGBM and XGBoost. The results of the remaining model pairs strongly support the alternative hypothesis. This implies that the performance difference of both LightGBM and XGBoost with their other competitors can be stated as significant. As a result, it can be stated that the best performing tree-based methods (LightGBM and XGBoost) achieve a significantly higher score than the best deep learning method (TabNet).

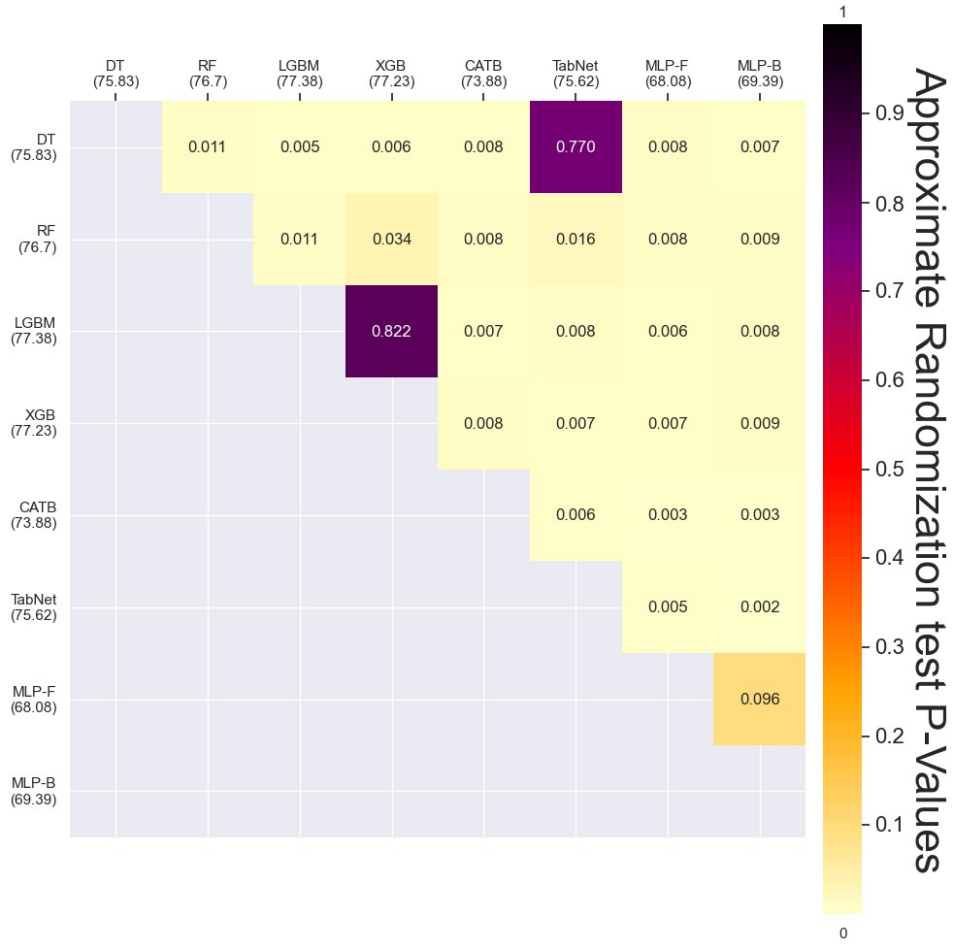

Figure 7: Model statistical comparison matrix

4.5 *Feature selection*

Next, the various feature importances are plotted that proved to contribute the most to correctly classifying a damage grade according to permutation, as described in section 4.5.1. For a description of each feature, please refer to Appendix A.

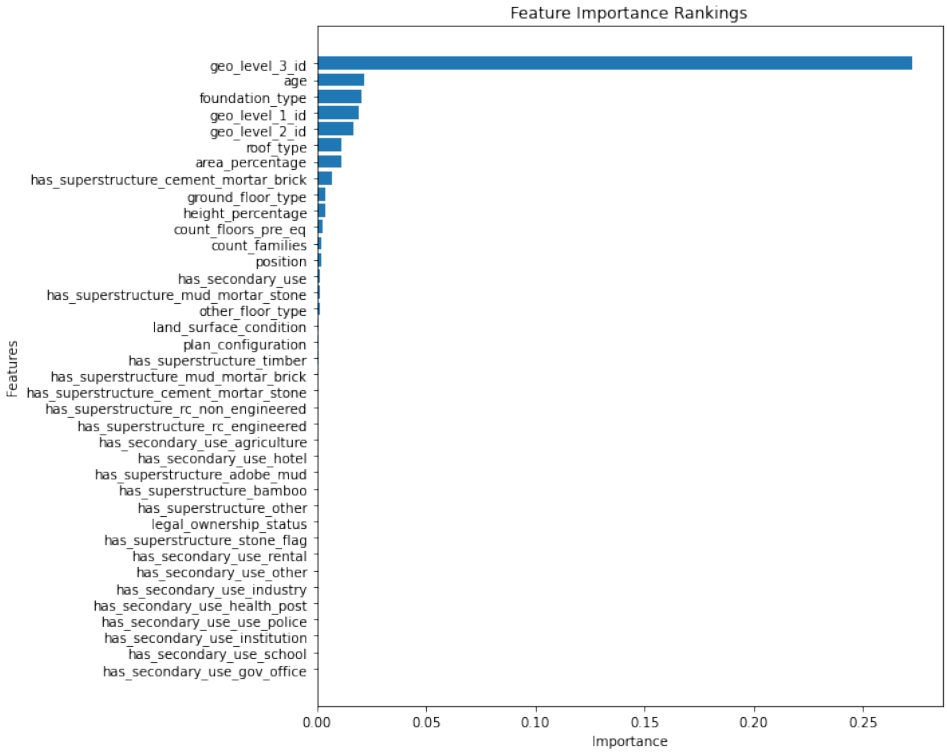

Figure 8: Feature importance by permutation

Unsurprisingly, the most important feature, *geo level 3 id*, has a relatively high importance score of 0.272842. This suggests that the specific geographic location at level 3 has a significant influence on the damage to buildings. The *age* of the buildings also demonstrates notable importance with a score of 0.021570, indicating that older buildings are more likely to be affected by the earthquake. Other important features include *foundation type*, *geo level 1 id*, and *geo level 2 id*, with importance scores of 0.020630, 0.019300, and 0.016602, respectively. These features provide insights into the type of foundation, and the broader geographic location at level 1 and level 2, which play obvious but crucial roles in determining the damage extent.

The remaining features, while having lower importance scores, still contribute to the prediction model to varying degrees. On the other hand, from the performance results of the model including and excluding feature selection, it's seen that for some models classification capabilities are incrementally enhanced by leaving out the least informative features. Low ranking features with permutation scores under 0.001, such as *land surface condition*, *plan configuration*, *has superstructure*, and *has secondary use* exhibited extremely low importance scores that have minimal impact

on predicting the damage to buildings. Therefore these features were selected to be excluded. Removing the least important features also helps reduce the risk of overfitting, where the model becomes overly specialized to the noise in the training data. By eliminating these less informative features, the model becomes more generalized and better able to capture the underlying patterns and relationships (Ghasemian et al., 2018).

5 DISCUSSION

This study aims to give insight into the capabilities of deep learning to outperform the conventional and parsimonious tree-based methods of building damage classification by earthquake in Nepal. By utilizing a large dataset of 50,001 instances, the efficacy of deep learning methods is tested for improved performance.

5.1 *Results discussion*

5.1.1 *Tree-based methods*

In assessing the predicament of method superiority, first, the tree-based methods are analyzed to set the benchmark for the deep learning methods to surpass. Between Decision Trees, Random Forests, LightGBM, XGBoost, and Catboost, both LightGBM and XGBoost proved to be the superior candidate tree-based method to set the benchmark in this study. This finding is in line with the results from the study of Borisov et al. (2022), where similarly, both of these models were concluded to be most effective amongst other (tree- and non tree-based) methods. As previously stated, the performance differences of these two models are statistically different from their less effective tree-based alternatives. However, since they do not significantly differ from each other, both models can be categorized as the most appropriate tree-based methods for the classification task of this dataset. To be able to fairly assess optimized TabNet and multi-layer perceptrons with tree-based methods, the same pre-processing steps are applied to the tree-based methods. The different pre-processing steps resulted in mixed effects on the outcomes of the tree-based models.

First, single pre-processing steps are applied to be able to evaluate the impact of each individual method. Notably, between these individual applications, feature selection has demonstrated its ability to slightly improve the base models of the decision trees and Catboost, although not substantial. For the cross-validation results, this resulted in a 0.11% (decision tree) and 0.01% (Catboost) improvement, respectively. For the decision tree this improvement did not substantially help generalize to the test data, due

to an increase of only 0.03%. Catboost, however, improved 0.16%. Even though the study of Ghasemian et al. (2018) found that tree-based models benefit from reducing dimensionality by feature selection, all tree-based candidates in this study did not benefit substantially.

Next, a data balancing technique are introduced to address class imbalance. However, it inadvertently compromises the models' performance. As all models decrease in performance, it turns out data balancing might not be a suitable solution. It enhances the models' capabilities to classify low and high damage grades, but by reducing its bias to classify the medium damage grade it performs worse on the original distribution. This is in line with findings of Johnson and Khoshgoftaar (2019), who showed that the potential of data balancing can substantially vary per domain and dataset.

Lastly, the combination of feature selection and data balancing is applied to explore the potential of a positive compounding effect of both steps. Unsurprisingly, this resulted in lower scores than applying solely feature selection, and higher scores than data balancing.

5.1.2 *Deep Learning methods*

When assessing the performance of the best configurations of the deep learning models, TabNet clearly outperforms both the multi-layer perceptron with fixed architecture and the multi-layer perceptron with a Bayesian optimized architecture. The specialized deep learning framework achieves micro-f1 scores that are significantly higher by 7.54% (MLP-Fixed) and 6.23% (MLP-Bayes) as displayed in the results section. This is contrasts the findings of Kadra et al. (2021) who showed that the MLP framework outperformed TabNet. Nevertheless, TabNet generalizes well to the test set, although a small drop of 0.78% is revealed.

Both the multi-layer perceptron methods, however, showcase an incremental yet noticeable rise in performance on the unseen data. Due to their relative poor performance, this might be caused by underfitting. As seen from the scores achieved by the other candidate models, there is substantial room for improvement on these methods.

When evaluating the efficacy of feature selection, no performance improvements are achieved across-the-board. Even though Grinsztajn et al. (2022) showed that uninformative features have a negative impact on deep learning, the removal of features with minimal permutation scores did not differ from scores achieved without removing uninformative features. Moreover, neither feature selection nor data balancing provide enhancements over the baseline configurations in all methods. Nevertheless, no substantial drops are showcased either. For feature selection this implies that despite the loss of volume of these features, the models are capturing the relevant relationships in the data.

To assess the overall discrimination ability of the best performing deep learning model, TabNet, the ROC-AUC is employed. It portrays TabNet's ability to achieve a high of true positive rate while maintaining a low false positive rate for damage grade 1, finalizing a score of 0.94. Damage grade 2, however, has a slightly lower score of 0.83. Since the data is imbalanced, the classifier is biased towards the majority class, which in turn also leads to high false positive rates which is the cause for this drop in performance. This is consistent with the findings of Gorishniy et al. (2021) who observed that TabNet results greatly varied amongst both skewed and non-skewed datasets. Damage grade 3 is distinguished gradually better, with a ROC-AUC of 0.89. These results are supported by the clear preference in prediction for damage grade 2 in the confusion matrix, thereby increasing false positives for this class. In addition, the f1-score of 50% from damage grade 1 portrays TabNet's lacking ability to correctly classify the minority class. This is caused by the high proportion of false positives in the precision and the false negatives in the recall within this class, considering its total support. Damage grades 2 and 3 perform considerably better in this aspect, achieving f1-scores of 79% and 73% respectively. TabNet's bias towards the dominantly represented classes is accountable for this.

5.1.3 *Tree-based methods versus deep learning*

The comparison between the highest scoring tree-based method, LightGBM, and the highest scoring deep learning method, TabNet, reveals moderately higher scores for LightGBM across all performance metrics. The tree-based method scores a 1.76% higher micro-f1 in cross-validation, accompanied by a standard deviation almost twice as low. This implies a more stable convergence which is supported by both the hyperparameter tuning plot in figure 6, and the observation from Grinsztajn et al. (2022). Both models show a similair slight drop in generalization capabilities to unseen data. The LightGBM ROC-AUC scores are only slightly better, indicating the capacity for slightly better discrimination between classes. The most substantial differentiating factor is the f1-score of damage grade 1. Both models struggle to minimize false positives and false negatives for this class, however, LightGBM scores 62% whereas TabNet scores an underwhelming 50%. This implies that LightGBM shows a relatively better capability in correctly identifying instances of damage grade 1 compared to TabNet. The performance difference is in line with the study of Borisov et al. (2022), who showed that competitiveness of deep learning becomes especially apparent in very large datasets with predominantly numerical features. The abundance of categorical features present in this dataset such as *roof type*, *foundation type*, and *ground floor type* amongst others therefore

may hamper the competitive potential of TabNet and its deep learning alternatives.

5.2 *Scientific and societal impact*

The clarity gained from comparing tree-based methods with the deep learning techniques of TabNet and multi-layer perceptrons can guide future researchers and practitioners in the field of earthquake damage assessment, enabling them to make informed decisions regarding the choice of appropriate machine learning techniques. This study shows that even though there is potential for deep learning techniques to become state-of-the-art according to Arik and Pfister (2021) and Kadra et al. (2021), this as not achieved by solely finding the optimal hyperparameters on the baseline configurations. Feature selection and data balancing techniques proved to not be robust enough to enhance the baselines for this task. Tree-based methods remain at the cutting-edge of performance for the earthquake classification domain, aligning with the results of Mangalathu et al. (2020) and Sajan et al. (2023).

Furthermore, this study contributes to the broader scientific community by expanding the knowledge base on the strengths and limitations of deep learning methods in the context of tabular data analysis. Ultimately, the societal impact of this research lies in providing insight into avenues that do not lead to better performance in classifying tabular earthquake building damage data. This knowledge expansion can inform decision-making processes on machine learning model selection for disaster response, renovation planning, and infrastructure development, leading to improved resilience and safety measures for communities at risk.

The different configurations of the deep learning models can be dissected, altered, or build upon to explore other paths that potentially improve upon the state-of-the-art benchmarks. Both in the earthquake building damage classification domain as well as the general tabular data domain.

5.3 *Limitations*

There are Limiting factors have to be taken into account. Firstly, due to computational and time restrictions, a downsized subsample of the dataset is utilized. Therefore, only 50,000 instances of the 260,000 total instances could be utilized. As the research of Grinsztajn et al. (2022) showed, deep learning models show potential to thrive on large volumes of data. As a consequence, the limited sample size may prove to result in substantially less competitive deep learning results. Secondly, only TabNet

was selected to assess the performance of a specialized deep learning framework to tabular data in this study. However, There is a variety of specialized deep learning frameworks that multiple studies deem capable of surpassing TabNet on tabular data, depending on the data at hand (Borisov et al., 2022). Thirdly, the work of Kadra et al. (2021) serves as the basis for the choice of multi-layer perceptron with its specific architecture setup and search space. However, not the full extend of his regularization techniques are applied in this study. Even though least prominent in the final best parameter result after hyperparameter tuning, methods such as Stochastic Weight Averaging, Lookahead Optimizer, Snapshot Ensembles, Skip Connections, and Adversarial learning extend the search space of his study. A fourth limitation of this study is the pre-processing selection. The pre-processing methods selected were chosen based on their efficacy on deep learning methods, as stated in the literature. For rigorous comparison, these methods are also applied to tree-based methods in order to evaluate their relative influence. Nonetheless, the performance of these tree-based methods could potentially benefit more from other pre-processing techniques, further disparaging the gap of results in this study. In terms of the feature selection itself, only one permutation score threshold is tested, where others could lead to different results. Lastly, the inherent feature importance method of TabNet was not utilized to gain more insight. Even though according to Bommert et al. (2020) the out-of-bag method for calculating permutation importance is robust, the attention mechanism of TabNet could prove to enrich the feature selection process.

5.4 *Future directions*

Applying the design of this study to other datasets within this domain could provide a more robust comparison of models. Besides broadening the variety of the data, raising the volume of the data is likely to have positive impact on the deep learning models (Grinsztajn et al., 2022). To further explore the potential enhancing pre-processing techniques, numerical embeddings by piece-wise linear encoding through quantile binning should be explored. According to the research by Gorishniy et al. (2022) this has the potential to substantially enhance deep learning model performance on tabular data specifically. Additionally, specialized deep learning frameworks of FT-Transformer, SAINT, and TabTransformer all showed competitive performances with both TabNet and tree-based methods and could therefore lead to state-of-the-art results (Borisov et al., 2022; Gorishniy et al., 2021; Grinsztajn et al., 2022). Future research incorporating the full scope of regularization techniques used by Kadra et al. (2021) could for-

tify multi-layer perceptrons as a challenger to specialized frameworks for most effective deep learning technique in the earthquake building damage classification domain.

6 CONCLUSION

In conclusion, this study adds to the existing body of research on the application of machine learning techniques for predicting building damage by earthquake using tabular data. It addresses the following research questions:

RQ1 *Which tree-based method offers the best performance?*

In assessing the results of the tree-based machine learning models, amongst a variety of configurations the baseline LightGBM proved to achieve the highest micro-f1 score compared with its alternatives. Superiority in both the cross-validation score, 77.38%, as well as the testing score 76,71%, ensuring its generalization capabilities. Upon statistical comparison, the difference between the baseline LightGBM and the baseline XGBoost revealed to be insignificant, therefore cementing both candidates as top performing tree-based methods in this study. Due to the class imbalance, classes having varying rates of correct classification.

RQ2 *Which deep learning framework proves to offer the best performance?*

The baseline TabNet showcased the best performance relative to the multi-layer perceptron candidates. It achieved the highest cross-validation score of 75.62%, significantly higher than its alternatives. Successful generalization capabilities are obtained resulting in a 74.84% test set score.

RQ3 *To what extent can pre-processing techniques impact the results of deep learning on the data?*

Both Feature selection and Data balancing techniques have proven insufficient in enhancing the baselines for the deep learning techniques on tabular data of building damage by earthquake.

MRQ *Does deep learning exhibit superior predictive power over Machine Learning tree-based models when classifying earthquake damage de- grees to buildings in Nepal?*

By answering these sub-questions, performance results of deep learning vs tree-based methods in this study collide with the results of deep learning

on tabular data by Borisov et al. (2022), Grinsztajn et al. (2022), and Shwartz-Ziv and Armon (2021). The research design of this study proved insufficient in providing evidence of superiority from deep learning methods over tree-based methods in classifying earthquake damage degrees to buildings in Nepal. Therefore, tree-based methods remain state-of-the-art in earthquake building damage classification, which is consistent with findings from Mangalathu et al. (2020) and Sajan et al. (2023).

7 REFERENCES

REFERENCES

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, *26*(10), 1340–1347. https://doi.org/10.1093/bioinformatics/btq134

Apolloni, B., & Ronchini, G. (1994). Dynamic sizing of multilayer perceptrons [Received 07 December 1992, Accepted 23 September 1993]. *Biol. Cybern.*, *71*(1), 49–63. https://doi.org/10.1007/BF00198911

Arik, S., & Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning.

Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics Data Analysis*, *143*, 106839. https://doi.org/https://doi.org/10.1016/j.csda.2019.106839

Borisov, V., Leemann, T., Sessler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2022). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. https://doi.org/10.1109/tnnls.2022.3229161

DrivenData. (2023). *Richter's Predictor: Modeling Earthquake Damage kernel description*. Retrieved March 10, 2023, from https://www.drivendata.org/competitions/57/nepal-earthquake/

Dzierżak, R. (2019). Comparison of the influence of standardization and normalization of data on the effectiveness of spongy tissue texture classification [Bibliography 26 references, figures, tables]. *Informatics, Automation, Measurements in Economy and Environmental Protection*, *9*(3), 66–69.

Feng, D.-C., Liu, Z.-T., Wang, X.-D., Chen, Y., Chang, J.-Q., Wei, D.-F., & Jiang, Z.-M. (2020). Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Construction and Building Materials*, *230*, 117000. https://doi.org/https://doi.org/10.1016/j.conbuildmat.2019.117000

Ghasemian, A., Hosseinmardi, H., & Clauset, A. (2018). Evaluating overfit and underfit in models of network community structure. *IEEE Transactions on Knowledge and Data Engineering*, *PP*. https://doi.org/10.1109/TKDE.2019.2911585

Ghimire, S., Guéguen, P., Giffard-Roisin, S., & Schorlemmer, D. (2022). Testing machine learning models for seismic damage prediction at a regional scale using building-damage dataset compiled after the 2015 gorkha nepal earthquake. *Earthquake Spectra*, *38*(4), 2970–2993. https://doi.org/10.1177/87552930221106495

Gorishniy, Y., Rubachev, I., & Babenko, A. (2022). On embeddings for numerical features in tabular deep learning.

Gorishniy, Y., Rubachev, I., Khrulkov, V., & Babenko, A. (2021). Revisiting deep learning models for tabular data.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data?

Hayes, A. F. (1998). Spss procedures for approximate randomization tests. *Behavior Research Methods, Instruments, & Computers*, *30*(3), 536–543. https://doi.org/10.3758/BF03200687

Huang, J., & Ling, C. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299–310. https://doi.org/10.1109/TKDE.2005.50

Hwang, S.-H., Mangalathu, S., Shin, J., & Jeon, J.-S. (2021). Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete building frames. *Journal of Building Engineering*, *34*, 101905. https://doi.org/https://doi.org/10.1016/j.jobe.2020.101905

Izenman, A. J. (2008). *Modern multivariate statistical techniques* (Vol. 1). Springer.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 27. https://doi.org/10.1186/s40537-019-0192-5

Kadra, A., Lindauer, M., Hutter, F., & Grabocka, J. (2021). Well-tuned simple nets excel on tabular datasets.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Kiani, J., Camp, C., & Pezeshk, S. (2019). On the application of machine learning techniques to derive seismic fragility curves. *Computers Structures*, *218*, 108–122. https://doi.org/https://doi.org/10.1016/j.compstruc.2019.03.004

King, R., Orhobor, O., & Taylor, C. (2021). Cross-validation is safe to use. *Nat Mach Intell*, *3*, 276. https://doi.org/10.1038/s42256-021-00332-z

Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize f1 score.

Lizundia, B., Davidson, R. A., Hashash, Y. M. A., & Olshansky, R. (2017). Overview of the 2015 gorkha, nepal, earthquake and the earthquake

spectra special issue. *Earthquake Spectra*, *33*(1_suppl), 1–20. https://doi.org/10.1193/120817eqs252m

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization.

Mangalathu, S., Sun, H., Nweke, C. C., Yi, Z., & Burton, H. V. (2020). Classifying earthquake damage to buildings using machine learning. *Earthquake Spectra*, *36*(1), 183–208. https://doi.org/10.1177/8755293019878137

Martins, L., & Silva, V. (2021). Development of a fragility and vulnerability model for global seismic risk analyses. *Bulletin of Earthquake Engineering*, *19*, 6719–6745. https://doi.org/10.1007/s10518-020-00885-1

Nateghi, R., Guikema, S. D., & Quiring, S. M. (2011). Comparison and validation of statistical methods for predicting power outage durations in the event of hurricanes. *Risk Analysis*, *31*(5), 705–719. https://doi.org/10.1111/j.1539-6924.2011.01618.x

Pargent, F., Pfisterer, F., Thomas, J., & Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features [Journal Article]. *Computational Statistics*, *37*(5), 2671–2692. https://doi.org/10.1007/s00180-022-01207-6

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

Raschka, S. (2020). Model evaluation, model selection, and algorithm selection in machine learning.

Sajan, S., Bhusal, A., Gautam, D., & Rupakhety, R. (2023). Earthquake damage and rehabilitation intervention prediction using machine learning. *Engineering Failure Analysis*, *144*. https://doi.org/10.1016/j.engfailanal.2022.106949

Sánchez-Maroño, N., Alonso-Betanzos, A., & Tombilla-Sanromán, M. (2007). Filter methods for feature selection–a comparative study. *Lecture notes in computer science*, *4881*, 178–187.

Sharma, K., Deng, L., & Noguez, C. C. (2016). Field investigation on the performance of building structures during the april 25, 2015, gorkha earthquake in nepal. *Engineering Structures*, *121*, 61–74. https://doi.org/10.1016/j.engstruct.2016.04.043

Shwartz-Ziv, R., & Armon, A. (2021). Tabular data: Deep learning is not all you need.

Siam, A., Ezzeldin, M., & El-Dakhakhni, W. (2019). Machine learning algorithms for structural performance classifications and predictions: Application to reinforced masonry shear walls. *Structures*, *22*, 252–265. https://doi.org/https://doi.org/10.1016/j.istruc.2019.06.017

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms.

Still, A. W., & White, A. P. (1981). The approximate randomization test as an alternative to the f test in analysis of variance. *British Journal of Mathematical and Statistical Psychology*, *34*(2), 243–252. https://doi.org/https://doi.org/10.1111/j.2044-8317.1981.tb00634.x

Trifunac, M., & Todorovska, M. (1997). Northridge, california, earthquake of 1994: Density of red-tagged buildings versus peak horizontal velocity and intensity of shaking. *Soil Dynamics and Earthquake Engineering*, *16*(3), 209–222. https://doi.org/https://doi.org/10.1016/S0267-7261(96)00043-7

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. *CoRR*, *abs/2104.10201*. https://arxiv.org/abs/2104.10201

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/https://doi.org/10.1016/j.neucom.2020.07.061

Ye, A., & Wang, Z. (2023). *Modern deep learning for tabular data: Novel approaches to common modeling problems* (1st ed.) [Published: 30 December 2022]. Apress. https://doi.org/10.1007/978-1-4842-8692-0

Yoon, K., Orhan, E., Kim, J., & Pitkow, X. (2021). Two-argument activation functions learn soft xor operations like cortical neurons.

Zimmer, L., Lindauer, M., & Hutter, F. (2021). Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl.

APPENDIX A

The dataset includes several features with their corresponding descriptions. These features are as follows:

Table 7: Full list of features

Feature	**Description**
geo_level_1_id, geo_level_2_id, geo_level_3_id (type: int)	Geographic region in which the building exists, from largest (level 1) to most specific sub-region (level 3). Possible values: level 1: 0-30, level 2: 0-1427, level 3: 0-12567.
count_floors_pre_eq (type: int)	Number of floors in the building before the earthquake.
age (type: int)	Age of the building in years.
area_percentage (type: int)	Normalized area of the building footprint.
height_percentage (type: int)	Normalized height of the building footprint.
land_surface_condition (type: categorical)	Surface condition of the land where the building was built.
foundation_type (type: categorical)	Type of foundation used while building.
roof_type (type: categorical)	Type of roof used while building.
ground_floor_type (type: categorical)	Type of the ground floor.
other_floor_type (type: categorical)	Type of constructions used in higher than the ground floors (except for roof).
position (type: categorical)	Position of the building.
plan_configuration (type: categorical)	Building plan configuration.
has_superstructure_adobe_mud (type: binary)	Flag variable indicating if the superstructure was made of Adobe/Mud.
has_superstructure_mud_mortar_stone (type: binary)	Flag variable indicating if the superstructure was made of Mud Mortar - Stone.
has_superstructure_stone_flag (type: binary)	Flag variable indicating if the superstructure was made of Stone.
has_superstructure_cement_mortar_stone (type: binary)	Flag variable indicating if the superstructure was made of Cement Mortar - Stone.

Feature	Description
has_superstructure_mud_ mortar_brick (type: binary)	Flag variable indicating if the superstructure was made of Mud Mortar - Brick.
has_superstructure_cement_ mortar_brick (type: binary)	Flag variable indicating if the superstructure was made of Cement Mortar - Brick.
has_superstructure_timber (type: binary)	Flag variable indicating if the superstructure was made of Timber.
has_superstructure_bamboo (type: binary)	Flag variable indicating if the superstructure was made of Bamboo.
has_superstructure_rc_non_ engineered (type: binary)	Flag variable indicating if the superstructure was made of non-engineered reinforced concrete.
has_superstructure_rc_ engineered (type: binary)	Flag variable indicating if the superstructure was made of engineered reinforced concrete.
has_superstructure_other (type: binary)	Flag variable indicating if the superstructure was made of any other material.
legal_ownership_status (type: categorical)	Legal ownership status of the land where the building was built.
count_families (type: int)	Number of families that live in the building.
has_secondary_use (type: binary)	Flag variable indicating if the building was used for any secondary purpose.
has_secondary_use_agriculture (type: binary)	Flag variable indicating if the building was used for agricultural purposes.
has_secondary_use_hotel (type: binary)	Flag variable indicating if the building was used as a hotel.
has_secondary_use_rental (type: binary)	Flag variable indicating if the building was used for rental purposes.
has_secondary_use_institution (type: binary)	Flag variable indicating if the building was used as a location of any institution.
has_secondary_use_school (type: binary)	Flag variable indicating if the building was used as a school.
has_secondary_use_industry (type: binary)	Flag variable indicating if the building was used for industrial purposes.
has_secondary_use_health_post (type: binary)	Flag variable indicating if the building was used as a health post.

Feature	Description
has_secondary_use_gov_office (type: binary)	Flag variable indicating if the building was used as a government office.
has_secondary_use_use_police (type: binary)	Flag variable indicating if the building was used as a police station.
has_secondary_use_other (type: binary)	Flag variable indicating if the building was secondarily used for other purposes.

APPENDIX B

Table 8: Search Space for Decision Tree Hyper Parameters

Parameter	Possible Values
`max_depth`	1–10 (integer)
`min_samples_split`	2–10 (integer)
`min_samples_leaf`	1–10 (integer)
`max_features`	`sqrt`, `log2`, `None`
`criterion`	`gini`, `entropy`
`min_impurity_decrease`	0.0–1.0 (real)
`class_weight`	`balanced`, `None`

Table 9: Search Space for Random Forest Hyper Parameters

Parameter	Possible Values
`n_estimators`	50–500 (integer)
`max_depth`	10, 20, 50, `None`
`min_samples_split`	2–10 (integer)
`min_samples_leaf`	1–4 (integer)
`max_features`	`sqrt`, `log2`, `None`
`bootstrap`	`True`, `False`

Table 10: Search Space for LightGBM Hyper Parameters

Parameter	Possible Values
learning_rate	0.001–1.0 (log-uniform)
n_estimators	100–500 (integer)
max_depth	-1–20 (integer)
num_leaves	20–80 (integer)
min_child_samples	10–30 (integer)
subsample	0.6–1.0 (uniform)
colsample_bytree	0.6–1.0 (uniform)
reg_alpha	0–0.5 (uniform)

Table 11: Search Space for XGBoost Hyper Parameters

Parameter	Possible Values
eta	0.01–1.0 (log-uniform)
lambda	1×10^{-10}–1.0 (log-uniform)
alpha	1×10^{-10}–1.0 (log-uniform)
num_round	50–500 (integer)
gamma	0.0–1.0
colsample_bylevel	0.1–1.0
colsample_bynode	0.1–1.0
colsample_bytree	0.1–1.0
max_depth	1–10 (integer)
max_delta_step	0–5 (integer)
min_child_weight	0.01–10.0 (log-uniform)
subsample	0.1–1.0

Table 12: Search Space for CatBoost Hyper Parameters

Parameter	Possible Values
learning_rate	1×10^{-5}–1.0 (log-uniform)
random_strength	1–20 (integer)
one_hot_max_size	0–25 (integer)
depth	0–10 (integer)
border_count	32–255 (integer)
l2_leaf_reg	1–10 (log-uniform)
bagging_temperature	0–1
leaf_estimation_iterations	1–10 (integer)

Table 13: Search Space for TabNet Hyper Parameters

Parameter	Possible Values
`Decision steps`	3-10 (uniform)
`Layer size`	8-128 (uniform)
`Relaxation factor`	1-2 (uniform)
`Lambda sparse`	0.0001-0.0 (uniform)
`Decay rate`	0.4-0.95 (uniform)
`Decay steps`	100-2000 (log-uniform)
`Learning rate`	0.0001-0.01 (uniform)

Table 14: Search Space for MLP-Fixed Hyper Parameters

Parameter	Possible Values
`dropout`	0.0–0.8 (uniform)
`l2reg`	1×10^{-5}–0.1 (log-uniform)
`batchnorm_active`	True, False

Table 15: Search Space for MLP-Bayes Hyper Parameters

Parameter	Possible Values
`n_hidden_layers`	1–16 (integer)
`n_neurons`	128–1024 (integer)
`l2reg`	1×10^{-5}–0.1 (log-uniform)
`dropout`	0.0–0.8 (uniform)
`batchnorm_active`	True, False