Impact of Categorical Feature Encoding on Machine Learning Based Shear Strength Prediction

WOUBISHET ZEWDU TAFFESE and GENDA CHEN

ABSTRACT

Recent advancements in machine learning (ML) offer powerful tools for predicting the structural integrity of civil infrastructure. A critical yet often overlooked aspect of ML modeling is data preprocessing, particularly categorical feature encoding. This study examines how different encoding schemes for interface conditions (monolithic, rough, smooth) impact ML predictions of interfacial shear strength. It compares categorical encodings (one-hot and five label encoding variations) with numerical friction coefficients (monolithic = 1.4, rough = 1.0, smooth = 0.6) across four ML models: eXtreme Gradient Boosting (XGBoost), Random Forest (RF), Support Vector Regression (SVR), and Artificial Neural Network (ANN).

Among 28 models, categorical encodings outperform numerical representations in 88% of cases, with ensemble models (RF, XGBoost) proving robust yet RF more sensitive to encoding variations. SVR and ANN exhibit encoding-dependent performance, with some SVR models achieving 12% higher accuracy than numerical-based models. Findings emphasize the crucial role of encoding choices in ML model performance, advocating for adaptive preprocessing techniques to enhance reliability in structural engineering and beyond.

INTRODUCTION

The rapid advancement of ML algorithms, coupled with increased computational power and the growing availability of structured datasets, has propelled the adoption of ML techniques in structural health monitoring. By leveraging data-driven methodologies, ML models can efficiently extract hidden patterns, enhance predictive accuracy, and optimize decision-making processes in structural engineering applications [1–3]. These datasets often comprise a mix of numerical, categorical, textual, and image-based information, reflecting the multifaceted challenges of real-world engineering problems. Among these data types, the integration of categorical and numerical features introduces unique complexities, particularly in how categorical data is represented and utilized within ML models.

Woubishet Z. Taffese and Genda Chen, Missouri University of Science and Technology, 500 W. 16th Street, Rolla, MO 65409, U.S.A.

Two of the most commonly used encoding techniques are label encoding and one-hot encoding [4, 5]. Label encoding is suited for categorical variables with an intrinsic ordinal relationship, as it assigns unique integer values to categories in a ranked order. However, this approach can inadvertently introduce unintended relationships, as ML models may interpret the numerical distances between encoded values as meaningful. Conversely, one-hot encoding ensures that each category is treated independently by generating separate binary columns, thereby eliminating any artificial ordinal assumptions.

This study systematically examines the impact of different encoding techniques on the predictive performance of four ML models, each operating on distinct principles, using a dataset from structural engineering. A key focus is placed on a categorical feature with an inherent ordinal relationship, which can also be represented through a corresponding numerical feature derived from experimental measurements—providing a unique opportunity to assess the efficacy of encoding choices. By evaluating various encoding schemes, this research aims to shed light on how data preprocessing influences model accuracy and reliability, offering insights into optimal encoding strategies for structural engineering datasets.

EMPLOYED ALGORITHMS

Machine learning regression models use different computational strategies to learn from data and make predictions. This study evaluates four distinct models—XGBoost, RF, ANN, and SVR—to assess how encoding techniques impact their performance.

XGBoost is a powerful gradient boosting algorithm that builds trees sequentially, where each new tree corrects the errors of the previous ones [6]. Unlike simple decision trees, it continuously updates the predictions by minimizing residual errors using gradient descent. XGBoost incorporates regularization techniques to prevent overfitting and uses parallel processing for faster computation. It is well-suited for structured data where feature interactions are important and provides high accuracy, especially when handling large datasets with complex dependencies.

RF, on the other hand, takes a different ensemble approach by training multiple decision trees independently on random subsets of the data (bootstrapping) and averaging their predictions [7]. This reduces variance and makes the model more robust against overfitting. Unlike XGBoost, which builds trees sequentially, RF constructs them in parallel, making it less sensitive to hyperparameters. While it performs well in high-dimensional datasets, it lacks the fine-tuned optimization process of XGBoost, making it slightly less efficient when working with highly complex interactions.

ANN takes a completely different approach by simulating the way human brains process information. Instead of relying on decision trees, ANN consists of layers of neurons that transform inputs through weighted connections and activation functions. It learns complex relationships by adjusting these weights using backpropagation and gradient descent [8]. Unlike tree-based models, ANN is excellent at capturing non-linear and high-dimensional patterns, making it particularly useful for applications like image processing and deep feature extraction. However, training an ANN requires substantial data, and it is computationally expensive, making it less efficient for small datasets compared to XGBoost or RF.

SVR adopts a fundamentally different principle based on finding a hyperplane that best fits the data within a defined margin [9]. Instead of minimizing squared errors like traditional regression models, SVR focuses on maintaining a balance between model complexity and prediction accuracy by using an epsilon-insensitive loss function. This allows it to ignore minor errors while capturing essential trends. Unlike XGBoost and RF, which rely on ensemble learning, or ANN, which uses neuron layers, SVR applies kernel tricks (such as radial basis functions) to map data into higher dimensions, making it highly effective for small datasets with complex relationships. However, it struggles with large datasets due to its high computational cost.

DATA AND METHODOLOGY

Experimental dataset

This study utilizes a refined dataset originally compiled by Edgmond and Sneed [10], focusing on a specific specimen type to ensure consistency and data completeness, resulting in a refined collection of 328 test samples. A detailed summary of the dataset's features, including units and key descriptive statistics for the numerical features, is presented in Table 1. A prominent categorical feature is the interface condition, which characterizes the surface state at the concrete-to-concrete interface. This feature is also represented numerically through experimentally derived friction coefficients (μ)—monolithic = 1.4, rough = 1.0, and smooth = 0.6—capturing their physical influence on shear strength as outlined by the ACI 318-14 guidelines. The remaining numerical features encompass various parameters that influence shear behavior, such as shear interface dimensions, concrete compressive strengths, shear reinforcement characteristics, normal stress, and interface bond capacity.

TABLE I. UTILIZED FEATURES

No	Feature	Unit	No	Feature	Unit
1	Interface condition (I)	[–]	6	Compressive strength, side 2 (f_{c2})	[MPa]
2	Coefficient of friction (μ)	[–]	7	Inclination degree (α)	[deg]
3	Length (L)	[mm]	8	Reinforcement ratio (ρ)	[%]
4	Width (W)	[mm]	9	Yield strength (f_y)	[MPa]
5	Compressive strength, side 1 (f_{c1})	[MPa]	10	Shear strength (τ)	[MPa]

Data encoding

Data encoding is one of the critical preprocessing steps for successful ML applications. In this study, the categorical feature representing the interface condition (I) is transformed into numerical formats using label and one-hot encoding. To explore the impact of encoding strategies, five distinct labeling approaches—L_1, L_2, L_3, L_4, and L_5—are applied, each designed to assess how different representations affect prediction accuracy.

Figure 1(a) illustrates the encoded values across these schemes. The first strategy, L_1, reverses the natural ordinal hierarchy by assigning higher values to smoother interfaces (3, 2, 1 for smooth, rough, and monolithic, respectively), enabling an

evaluation of how flipping the hierarchy impacts model performance. L_2 uses a uniform incremental approach, assigning values of 1, 2, and 3 to smooth, rough, and monolithic conditions, respectively, serving as a baseline for comparison.

L_3 examines the effect of reversing a non-uniform encoding, where smooth, rough, and monolithic conditions are encoded as 6, 2, and 0. This strategy explores the impact of an inverted hierarchy in a nonlinear context. L_4 employs a more pronounced non-uniform increment, encoding smooth, rough, and monolithic conditions as 0, 2, and 6, respectively, emphasizing bond strength differences. L_5 encodes the interface conditions based on the friction coefficients defined by ACI 318-14 (0, 1, and 1.4 for smooth, rough, and monolithic conditions, respectively), aligning with engineering practices to assess the practical relevance of encoding choices.

Figure 1(b) illustrates the transformation of the feature I using one-hot encoding. This method ensures that each category is treated independently, eliminating any assumptions of ordinal relationships between them. One-hot encoding generates binary columns for each category, where a value of 1 indicates the presence of a category and 0 denotes its absence, capturing the categorical nature without implying any hierarchy.

	L_1	L_2	L_3	L_4	L_5
Smooth	3	1	6	0	0
Rough	2	2	2	2	1
Monolithic	1	3	0	6	1.4

(a)

	Smooth	*Rough*	*Monolithic*
Smooth	1	0	0
Rough	0	1	0
Monolithic	0	0	1

(b)

Figure 1 Feature encoding methods: (a) Lable encoding and (b) One-hot encoding

Model training and evaluation encoding

After completing fundamental data preprocessing, including data normalization and partitioning, seven models were developed. Normalization, using min-max scaling, was applied to ANN and SVR as a standard practice. The dataset was randomly split into 80% training and 20% testing subsets to ensure a balanced evaluation of model performance.

Table 2 outlines the input features and encoding methods used for each model. Models M-1 through M-5 share identical input and target features, differing only in the labeling schemes (L_1–L_5) applied to the interface condition (I). M-6 employs the same input features but utilizes one-hot encoding for I. Model M-7, however, replaces the categorical interface condition with its numerical counterpart—the coefficient of friction (μ), eliminating the need for label encoding.

During model training, each algorithm undergoes rigorous hyperparameter optimization using a combination of grid search and k-fold cross-validation to enhance predictive performance. Among nine tested k-fold values (ranging from 2 to 10), 5-fold cross-validation consistently yielded the lowest mean square error (MSE) and was selected to ensure optimal model evaluation and stability.

All modeling procedures are implemented in Python using the scikit-learn library [11]. Model performance is assessed using four key statistical metrics: mean absolute error (MAE), MSE, root-mean-square error (RMSE), and coefficient of determination (R^2). For detailed descriptions of these metrics, refer to [12].

TABLE II. OVERVIEW FEATURES AND ENCODING METHODS APPLIED

Model	Input feature	Encoding type	Target feature
M-1	$I, L, W, f_{c1}, f_{c2}, \alpha, \rho, f_y, \sigma_N$	Label L_1	τ
M-2		Label L_2	
M-3		Label L_3	
M-4		Label L_4	
M-5		Label L_5	
M-6		One-hot	
M-7	$\mu, L, W, f_{c1}, f_{c2}, \alpha, \rho, f_y, \sigma_N$	-	

RESULTS AND DISCUSSION

Figure 2 presents the percentage change in model performance—evaluated using MAE, RMSE, and R^2—relative to M-7, where interface conditions are represented using numerical friction coefficients. Subfigures 2(a–c) compare the performance of M-1 through M-5 across the three metrics, while 2(d) is dedicated to M-6. The findings reveal that 85% of models trained with label encoding and all models trained with one-hot encoding achieve higher R^2 values than those using numerical friction coefficients.

Among standalone models, both SVR and ANN exhibit pronounced sensitivity to labeling schemes. While certain SVR models attain higher R^2 alongside reduced MAE and RMSE compared to numerical-based models, half of the ANN models experience a decline in R^2. Notably, an SVR model employing label encoding that disrupts ordinal relationships achieves a remarkable 12% improvement in accuracy over its M-7.

Ensemble models demonstrate superior robustness to variations in encoding strategies. Among them, XGBoost stands out as the most robust, consistently maintaining stable performance across MAE, RMSE, and R^2 metrics regardless of the encoding scheme. RF also benefits significantly from categorical encoding, achieving lower MAE and RMSE compared to M-7. In particular, under the L_4 labeling scheme with one-hot encoding, RF's R^2 improves by more than 6% over M-7. These results underscore the advantages of ensemble methods when integrated with categorical encoding, reinforcing their effectiveness in predictive modeling.

Figure 3 compares the residual distributions of M-7, where μ directly represents interface conditions, with the best-performing model for each algorithm. In XGBoost (Figure 3(a)), all models (M-1 through M-6) utilizing categorical encodings yield residual distributions nearly indistinguishable from M-7, reflecting the model's encoding-agnostic robustness. In RF (Figure 3(b)), M-6, which employs one-hot encoding, exhibits a narrower interquartile range (IQR) and shorter whiskers than M-7, signaling improved predictive stability and reduced error variance.

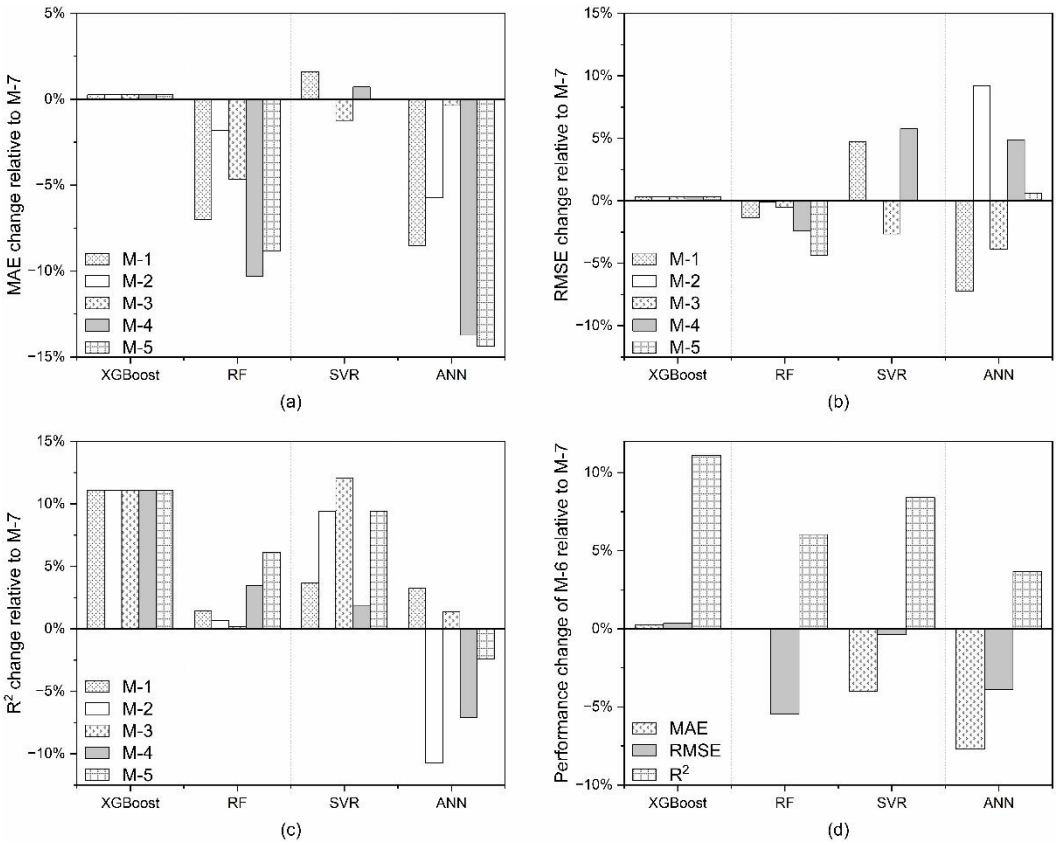

Figure 2 Performance comparison of models relative to M-7: (a–c) M-1 to M-5 evaluated by MAE, RMSE, and R^2; (d) M-6 evaluated by all three metrics.

In Figure 3(c), the residual distribution for SVR reveals that M-3 exhibits shorter whiskers than M-7, indicating reduced prediction variance. M-3 adopts a reverse ordinal label encoding, where "smooth", "rough", and "monolithic" are encoded as 6, 2, and 0, respectively. This configuration imposes a non-linear and exaggerated scaling across interface types. Similarly, in the case of ANN, M-1 demonstrates a smaller interquartile range (IQR) and shorter whiskers than M-7, reflecting enhanced performance. M-1 also employs a non-ordinal encoding—"smooth" as 3, "rough" as 2, and "monolithic" as 0—but with a narrower spread between categories compared to M-3.

Among label-encoded models, M-2, M-4, and M-5 preserve ordinal relationships but differ in scaling. M-5 aligns precisely with the experimental μ values, reflecting direct numerical observations. While maintaining ordinal structures is generally advocated to minimize model confusion, none of these models outperform those employing alternative encoding strategies. Instead, RF achieves its best performance using one-hot encoding—an approach typically reserved for categorical variables without inherent order. Moreover, the underperformance of M-7, which adheres to conventional engineering practice, suggests that representing interface conditions solely through numerical friction coefficients may not be the optimal approach for ML predictive modeling.

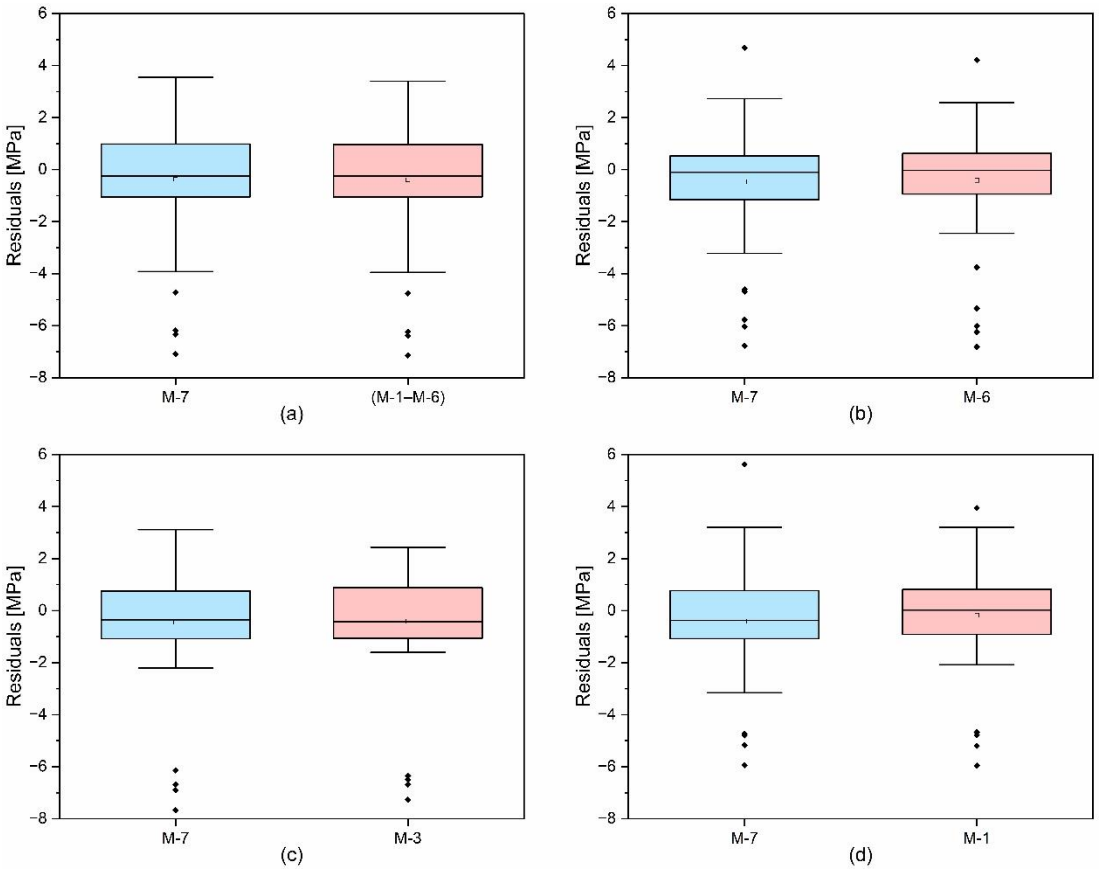

Figure 3 Residuals comparison of M-7 against the best-performing models: (a) XGBoost, (b) RF, (c) SVR, and (d) ANN.

The observed paradox—where categorical feature encoding improves shear strength prediction accuracy despite violating ordinal relationships—calls for flexible and adaptive encoding strategies that balance procedural soundness with predictive accuracy. Furthermore, since the dataset used in this analysis originates from a single structural engineering domain, broader investigations are warranted. Future work will explore the impact of encoding strategies across diverse datasets and algorithmic frameworks, providing deeper insights into their role in predictive modeling across various structural engineering applications and beyond.

CONCLUSIONS

This study explored the impact of encoding strategies on ML model performance in predicting shear resistance at concrete-to-concrete interfaces. Results demonstrated that encoding choices profoundly affect predictive accuracy, particularly in standalone models. Notably, an SVR model that disrupted ordinal relationships outperformed numerical friction coefficient-based models by 12%, highlighting the critical role of encoding in ML applications.

A key paradox emerged in the trade-off between procedural clarity and predictive accuracy. While encoding schemes grounded in engineering principles maintained

methodological rigor, alternative labeling schemes—despite being less conventional—often delivered superior model performance. For instance, RF achieved its best results using one-hot encoding, a method typically reserved for non-ordinal categories, further underscoring the complexity of encoding decisions in predictive modeling.

Ensemble models, particularly XGBoost, exhibited remarkable robustness to encoding variations, consistently outperforming standalone models. However, despite the ordinal nature of the categorical feature, ML models did not uniformly benefit from preserving these relationships, leading to unexpected performance trends. These findings highlight the need for adaptive encoding strategies that balance predictive accuracy with procedural soundness. Future research should explore optimized feature representations across diverse datasets and algorithms to refine ML-driven structural analysis and enhance its generalizability.

REFERENCES

1. Hu T, Zhang H, Khodadadi N, Taffese WZ, Nanni A. Enhancing bond strength prediction at UHPC-NC interface: A data-driven approach with augmentation and explainability. *Constr Build Mater*. 2024;451:138757. doi:10.1016/j.conbuildmat.2024.138757
2. Cheng C, Taffese WZ, Hu T. Accurate Prediction of Punching Shear Strength of Steel Fiber-Reinforced Concrete Slabs: A Machine Learning Approach with Data Augmentation and Explainability. *Buildings*. 2024;14(5):1223. doi:10.3390/buildings14051223
3. Zhu Y, Taffese WZ, Chen G. Enhancing FRP-concrete interface bearing capacity prediction with explainable machine learning: A feature engineering approach and SHAP analysis. *Eng Struct*. 2024;319:118831. doi:10.1016/j.engstruct.2024.118831
4. Muthu S. *Machine Learning Optimization Strategies for High Performance Cyber Data Analytics*. RK Publication; 2024.
5. Cuantum Technologies. *Machine Learning Hero: Master Data Science with Python Essentials: Machine Learning with Python Hands-On Guide from Beginner to Expert*. Vol 1.; 2024.
6. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ; 2016:785-794. doi:10.1145/2939672.2939785
7. Breiman L. Bagging predictors. *Mach Learn*. 1996; 24:123-140. doi:https://doi.org/10.1007/BF00058655
8. Haykin S. *Neural Networks and Learning Machines*. 3rd ed. Pearson Education, Inc.; 2009.
9. Shalev-Shwartz S, Ben-David S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press; 2014.
10. Edgmond NJ, Sneed LH. *Examination of Shear Friction Design Provisions.*; 2018.
11. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011;12(85):2825-2830.
12. Taffese WZ, Zhu Y, Chen G. Explainable AI based slip prediction of steel-UHPC interface connected by shear studs. *Expert Syst Appl*. 2025;259:125293. doi:10.1016/j.eswa.2024.125293