**Predictive Modeling and Analysis of 28-Day Compressive Strength Using Artificial Neural Networks**

**1. Introduction**

Concrete compressive strength is one of the most fundamental and widely used mechanical properties for assessing structural integrity and durability in civil engineering. Accurate prediction of this property is essential for quality control, design optimization, and sustainability in construction. Typically, compressive strength is measured at intervals such as 7, 14, 21, and 28 days after casting to monitor strength development. This study aims to build a robust predictive model using Artificial Neural Networks (ANN) to estimate the 28-day compressive strength based on early-age compressive strength data and sample types. The analysis includes data preprocessing, exploratory analysis, correlation examination, and model evaluation.

**2. Dataset Overview**

The dataset contains **180 samples**, each with measured compressive loads and compressive strengths at 7, 14, 21, and 28 days, along with the type of material or sample used.

**2.1 Dataset Description**

* **No missing values** were observed.
* **Compressive load values** were dropped due to extremely high correlation (r > 0.9999) with corresponding strength values.
* **Final dataset** contains compressive strength values (at 4 stages) and encoded sample type features.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| 7D, 14D, 21D, 28D | Compressive strength (N/mm²) |
| Sample Type | Categorical variable (e.g., Control, 0.5%H, etc.) |

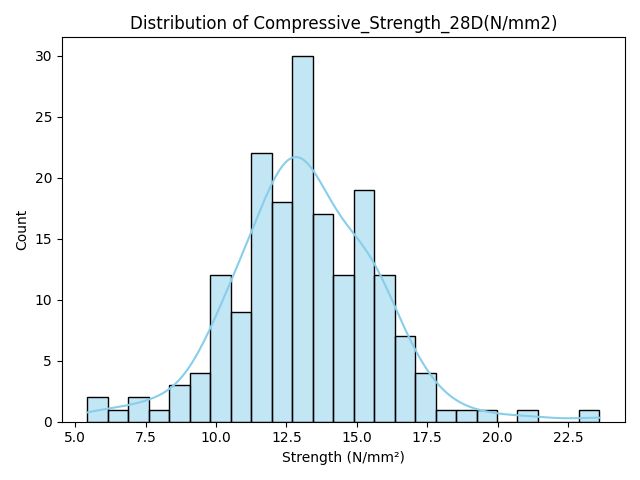
**2.2 Statistical Summary of Compressive Strength(N/mm²)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **7 Days** | **14 Days** | **21 Days** | **28 Days** |
| Count | 180 | 180 | 180 | 180 |
| Mean | 12.32 | 14.47 | 11.90 | 13.19 |
| Std. Dev. | 2.29 | 2.56 | 2.05 | 2.60 |
| Min | 5.12 | 8.14 | 5.65 | 5.44 |
| 25th Percentile | 10.93 | 13.10 | 10.42 | 11.63 |
| Median | 12.30 | 14.37 | 11.86 | 13.09 |
| 75th Percentile | 13.72 | 15.37 | 13.31 | 15.04 |
| Max | 19.52 | 28.52 | 17.39 | 23.61 |

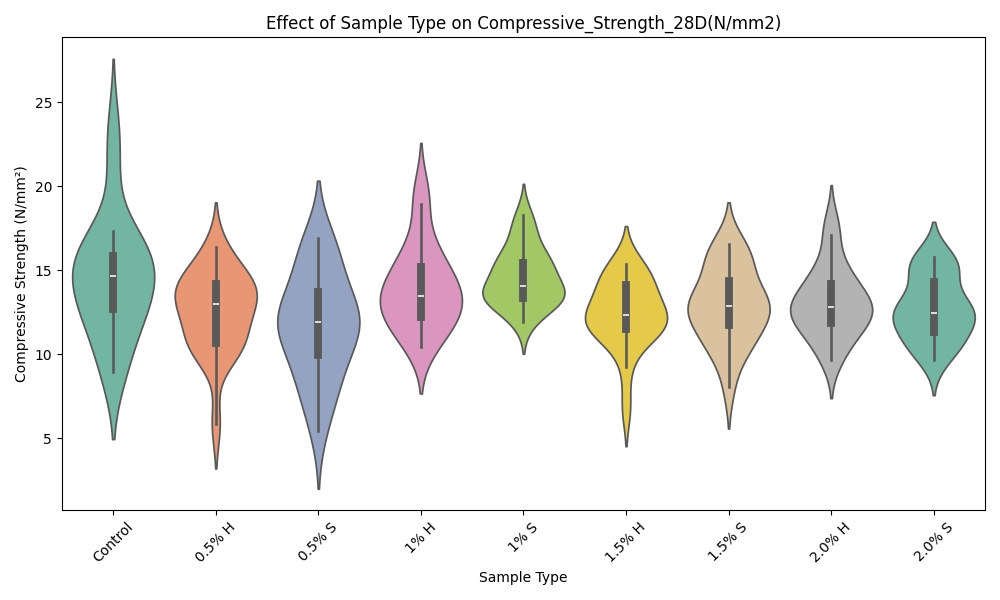
**3. Exploratory Data Analysis (EDA)**  
EDA was instrumental in understanding the distribution and variability of tensile strength values:

* **Histograms**: Compressive strength at all time points approximated a normal distribution, with the 28-day samples fall within 10–17.5 N/mm².
* **Boxplots**: These visualizations revealed outliers, particularly at 21 and 28 days, indicating potential variability in the late-age measurements.
* **Violin Plots**: 28-day tensile strength was plotted by sample type to assess intra-group spread and density.
* **Correlation Matrix**: A Pearson correlation map was generated to quantify the linear relationship between tensile strength values across all time intervals.

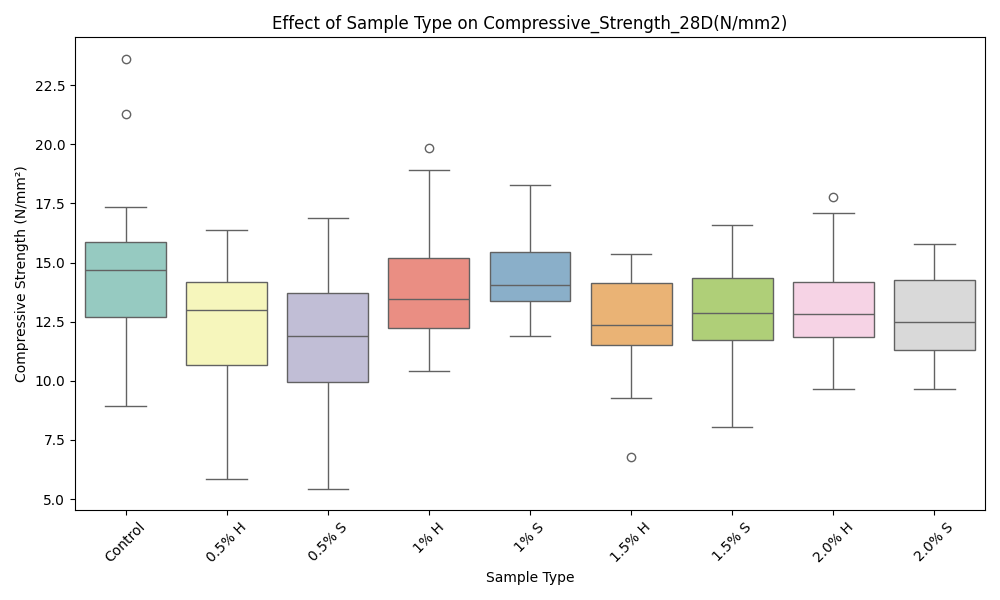
*Note*: While plots were generated for all time points (7D–28D), this report includes only 28-day visuals to maintain clarity and focus on the prediction target.

****

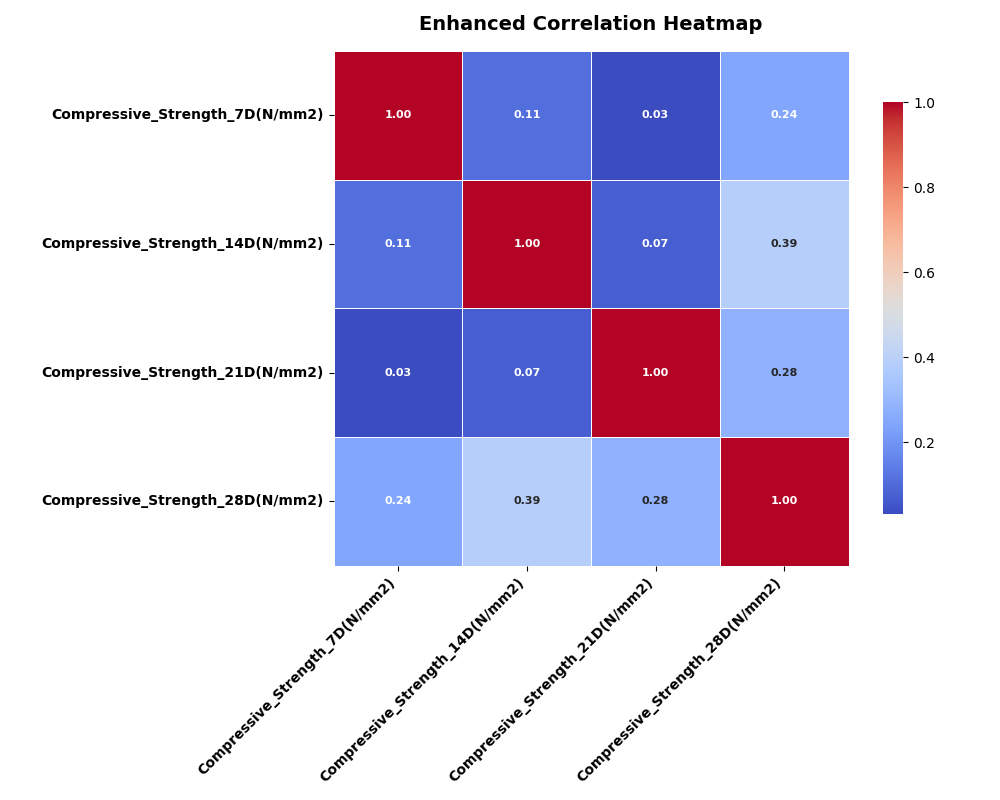
**Figure 1**: Histogram of 28-Day Tensile Strength

****

**Figure 2**: Violin Plot of 28-Day Strength by Sample Type



**Figure 3**: Boxplot of 28-Day Strength (Outliers Highlighted)



**Figure 4**: Correlation Heatmap of Tensile Strength Measurements

**4. Data Preprocessing**

To prepare the data for modeling:

* **Reordering Columns**: Columns were reordered to prioritize *Sample Type* and group load and strength measurements.
* **Handling Multicollinearity**: Tensile load and strength features showed near-perfect correlations (≈0.9999) (Kutner et al., 2005), leading to the removal of load columns to avoid multicollinearity, which can destabilize models (Hair et al., 2019).
* **One-Hot Encoding**: The categorical *Sample Type* was encoded into dummy variables (e.g., *Sample Type\_Control*).
* **Outlier Handling**: Numeric features were capped using the Interquartile Range (IQR) method (1.5 × IQR bounds).
* **Scaling**: Features were standardized using *RobustScaler* to mitigate outlier effects.
* **Data Splitting**: The dataset was split into 80% training (144 samples) and 20% testing (36 samples) sets, with 14 features after preprocessing.

**5. Model Architecture and Training**

An Artificial Neural Network (ANN) was built using Keras with the following configuration:

* **Input Layer**: 14 nodes
* **Hidden Layers**: Two dense layers with 32 units each, ReLU activation
* **Regularization**: Dropout (0.2) + BatchNormalization
* **Optimizer**: Adam (lr=0.001)
* **Loss Function**: Mean Squared Error (MSE)
* **Callbacks**: EarlyStopping with patience of 15 on validation loss

Training was conducted for up to 200 epochs with 10% of the training set used for validation.

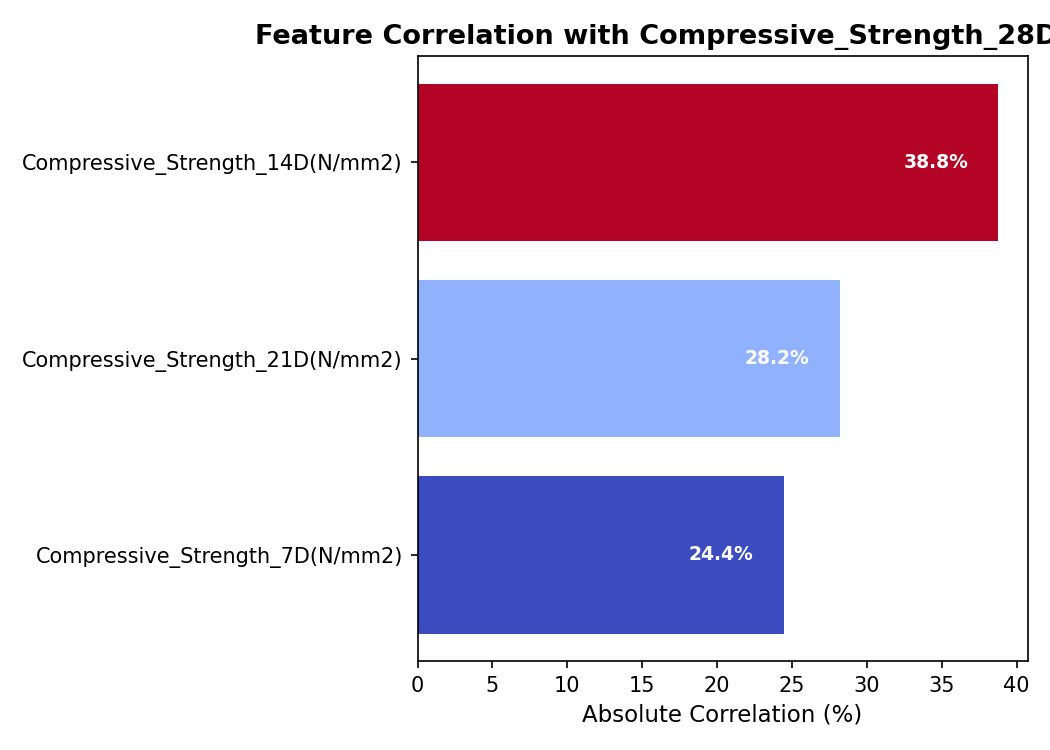
**6. Model Evaluation**

Performance on the test set:

* **MSE**: 0.4855
* **R²**: 0.7885

This indicates that approximately 79% of the variance in 28-day compressive strength is explained by the model.

**Feature Importance**  
Permutation-based feature importance identified 14-day tensile strength as the most influential predictor. Earlier time points and sample type features also contributed but to a lesser extent.



**7. Interpretation and Discussion**

The ANN model demonstrates a strong ability to generalize predictions from early compressive strength data and sample type features.

* **Predictive Strength of Features**: The 14-day and 21-day strengths are the most influential predictors for 28-day strength.
* **Categorical Influence**: Sample Type added value to prediction, suggesting differences in mix design or composition.
* **Model Limitations**: Slight underfitting due to high variance among later-age strengths. Including concrete mix proportions or curing conditions may improve results.

**8. Conclusion**

This study validates the efficacy of ANNs in predicting 28-day compressive strength using early-age data and sample types. The methodology balances accuracy and interpretability through careful preprocessing and model tuning. Future research could integrate additional concrete properties such as cement content, water-cement ratio, and admixtures for further model enhancement.

**References**

* Chollet, F. (2015). *Keras: The Python Deep Learning library*. [https://keras.io](https://keras.io/)
* Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
* Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
* Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
* Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.