

Proceeding Paper

# A Clustering-Enhanced Explainable Approach Involving Convolutional Neural Networks for Predicting the Compressive Strength of Lightweight Aggregate Concrete <sup>†</sup>

Violeta Migallón , Héctor Penadés \*  and José Penadés 

Department of Computer Science and Artificial Intelligence, University of Alicante, 03690 Alicante, Spain; violeta@ua.es (V.M.); jpenades@ua.es (J.P.)

\* Correspondence: jhector.penades@ua.es

<sup>†</sup> Presented at the 6th International Electronic Conference on Applied Sciences, 9–11 December 2025; <https://sciforum.net/event/asec2025>.

## Abstract

Lightweight aggregate concrete (LWAC) is a practical alternative to conventional concrete in civil engineering, offering advantages such as reduced density, enhanced insulation properties, and improved seismic performance. However, segregation during compaction remains a limitation, as it can lead to non-uniform material distribution and reduced compressive strength. This study addresses this issue by combining non-destructive techniques with deep learning methods to predict the compressive strength of LWAC. We propose an explainable approach based on a convolutional recurrent neural network architecture, enhanced by unsupervised clustering and SHapley Additive exPlanations (SHAP), to improve interpretability. To optimize predictive performance, several aggregation strategies are evaluated at the recurrent layer before the dense layers, including full-sequence flattening, max pooling, average pooling, and an attention mechanism over the full sequence. Experimental results show that the proposed model outperforms conventional machine learning methods such as multilayer perceptron (MLP), random forest (RF), and support vector regression (SVR), as well as ensemble methods such as gradient boosting (GBR), XGBoost, and weighted average ensemble (WAE). Furthermore, when combined with unsupervised clustering, the model identifies latent behavioral patterns that are not observable through traditional evaluation techniques. This demonstrates the potential of integrating non-destructive testing with interpretable deep learning as a reliable approach for the structural assessment of LWAC.

**Keywords:** lightweight aggregate concrete; compressive strength prediction; explainable artificial intelligence; deep learning models



Academic Editor: Eugenio Vocaturro

Published: 11 March 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Concrete is one of the most widely used materials in civil engineering. However, current construction demands, such as the need for lightweight and high-strength structures, have led to the increased use of lightweight aggregate concrete (LWAC). This type of concrete offers several advantages, including low density, improved thermal and acoustic insulation, and enhanced structural efficiency [1]. These properties help reduce structural loads and improve the seismic performance of buildings [1]. Despite these benefits, one of the main issues associated with LWAC is segregation. During compaction, lightweight aggregates (LWAs) can rise to the surface, while denser particles settle at the bottom,

leading to a loss of uniformity and a significant reduction in compressive strength. This segregation creates zones with varying composition and porosity, which compromise the structural integrity and durability of the element. Contributing factors include particle size and density, the materials used, and the compaction method applied. Therefore, it is essential to control the behavior of LWAC in the fresh state by adjusting its rheology to ensure workability and by applying sufficient vibration to remove air without causing segregation [2].

Given the difficulty of controlling segregation with conventional methods, predictive modeling offers a promising complementary approach for evaluating LWAC performance. Machine learning is increasingly applied in concrete research due to its ability to model complex and nonlinear relationships [3,4]. It has proven especially useful in predictive engineering tasks, such as modeling the compressive strength of LWAC. The wide variability in LWAC mix designs, along with the inherent complexity of cementitious systems, makes them well-suited to machine learning techniques, which provide versatile and accurate modeling capabilities [5–10]. Among the most commonly used machine learning techniques for predicting concrete properties are artificial neural networks (ANNs), support vector regression (SVR), regression trees (RT), random forest (RF), gradient boosting regression (GBR), and K-nearest neighbors (KNN) [5,11–15].

Recent research has also investigated deep learning methods. In [16], a fully connected feed-forward network with four hidden dense layers was proposed to predict concrete compressive strength from destructive and non-destructive tests. In [17], a hybrid deep learning model was introduced that combines convolutional, attention, residual, and fusion modules to predict compressive strength using a dedicated ultrasonic signal dataset. Although these contributions demonstrate the potential of advanced learning methods, the combined use of hybrid deep architectures, interpretability, and unsupervised analysis remains underexplored in the context of LWAC and non-destructive parameters.

The main contribution of this study is the development of an interpretable hybrid deep learning model that predicts the compressive strength of LWAC from ultrasonic pulse velocity (UPV) and mix-related parameters, combining explainability with unsupervised analysis to support non-destructive quality assessment. Numerical experiments show that the convolutional gated recurrent neural networks developed and fine-tuned in this work outperform other machine learning approaches and recent studies in this area [5,10]. Furthermore, interpretability techniques and unsupervised clustering are applied to enhance the analysis of predictions, providing insights into influential features and uncovering patterns that standard evaluation metrics may not capture.

## 2. Materials and Methods

### 2.1. Description and Exploratory Analysis of the Dataset

The experimental program, as described in [5], focused on LWAC produced using four mixture designs, combining two target densities (1700 and 1900 kg/m<sup>3</sup>) and two aggregate types with different particle densities (482 and 1019 kg/m<sup>3</sup>). The mixes were defined using the Fanjul method [18], and segregation was intentionally induced by varying the vibration time. A segregation index was computed by dividing the local P-wave velocity of each core segment by the mean velocity of the four segments of the corresponding specimen, thereby quantifying deviations from uniformity. Each core segment was characterized by eight variables or features, including mixture parameters, compaction conditions, density, ultrasonic measurements, and compressive strength, resulting in a dataset of 640 instances (see Table 1).

**Table 1.** Summary statistics of the experimental dataset.

Variable	Minimum	Maximum	Mean $\pm$ SD	CV (%)
LWAC fixed density (kg/m <sup>3</sup> )	1700	1900	1800.00 $\pm$ 100.08	5.56
LWA particle density (kg/m <sup>3</sup> )	482	1019	750.50 $\pm$ 268.71	35.80
Concrete laying time (min)	15	90	48.75 $\pm$ 28.83	59.14
Vibration time (s)	0	80	30.00 $\pm$ 28.31	94.37
Experimental dry density (kg/m <sup>3</sup> )	1069.80	2486.84	1673.35 $\pm$ 179.14	10.71
P-wave velocity (m/s)	3044.25	5253.73	3778.89 $\pm$ 370.88	9.81
Segregation index	0.845	1.136	1.000 $\pm$ 0.0352	3.52
Compressive strength (MPa)	2.99	50.72	21.55 $\pm$ 8.97	41.62

SD = standard deviation; CV = coefficient of variation; LWAC = lightweight aggregate concrete; LWA = lightweight aggregate.

The strongest correlations were those involving compressive strength. Specifically, LWA particle density ( $\rho = 0.708$ ) and experimental dry density ( $\rho = 0.649$ ) showed the strongest positive correlation with compressive strength, as both are directly associated with the physical structure and compactness of the concrete. Additional relevant positive correlations were observed, such as between LWAC fixed density and both the experimental dry density ( $\rho = 0.646$ ) and P-wave velocity ( $\rho = 0.520$ ) (see Table 2).

**Table 2.** Spearman correlation matrix of the dataset variables.

	LWAC-FD	LWA-PD	CLT	VT	DD	PWV	SI	CS
<b>LWAC-FD</b>	1.000	0.000	0.000	0.000	0.646	0.520	-0.023	0.295
<b>LWA-PD</b>	0.000	1.000	0.000	0.000	0.170	-0.402	-0.012	0.708
<b>CLT</b>	0.000	0.000	1.000	0.000	-0.019	0.150	0.002	0.046
<b>VT</b>	0.000	0.000	0.000	1.000	0.082	0.122	0.014	0.071
<b>DD</b>	0.646	0.170	-0.019	0.082	1.000	0.482	0.295	0.649
<b>PWV</b>	0.520	-0.402	0.150	0.122	0.482	1.000	0.378	0.014
<b>SI</b>	-0.023	-0.012	0.002	0.014	0.295	0.378	1.000	0.184
<b>CS</b>	0.295	0.708	0.046	0.071	0.649	0.014	0.184	1.000

LWAC-FD = LWAC fixed density; LWA-PD = LWA particle density; CLT = Concrete laying time; VT = Vibration time; DD = Experimental dry density; PWV = P-wave velocity; SI = Segregation index; CS = Compressive strength.

## 2.2. Model Architecture Design

We developed a hybrid deep learning architecture that combines convolutional and recurrent layers to predict the compressive strength of LWAC. Multiple configurations for each layer type were explored, including convolutional filters, recurrent units, dense layer sizes, and activation functions. The final model architecture was selected based on Monte Carlo cross-validation, and it is summarized in Table 3.

**Table 3.** Summary of the model architecture.

Layer (Type)	Output Shape	Param. #
Input Layer	(B, 7, 1)	0
Conv1D (filters = 32, kernel = 3)	(B, 5, 32)	128
GRU (units = 12, return_sequences)	(B, 5, 12)	1656
Flatten	(B, 60)	0
Dense 1 (units = 36)	(B, 36)	2196
Dense 2 (units = 18)	(B, 18)	666
Dense 3 (units = 9)	(B, 9)	171
Dense Output (units = 1)	(B, 1)	10
Total		4827

B = batch size; Conv1D = 1D convolution layer; GRU = Gated Recurrent Unit; Dense = fully connected layer.

Each input sample is represented as a one-dimensional vector of seven features. Features were scaled using Min–Max normalization to the range [0, 1], with scaling parameters calculated on the training data. A 1D convolutional layer with 32 filters is then applied, where each filter slides a kernel of size three along the input sequence (stride = 1, no padding). At each position, the kernel performs element-wise multiplication with the corresponding local region of the input and sums the results, generating feature maps that capture local interactions between neighboring features. This operation reduces the sequence length from seven to five. In our architecture, this layer uses ReLU activation and He-uniform initialization for the weights. The convolved output is processed by a Gated Recurrent Unit (GRU) layer [19], a simplified type of recurrent neural network that merges the cell and hidden states into a single hidden state and uses two gating mechanisms: the update gate and the reset gate. These gates regulate the flow of information through the network, enabling it to selectively retain important features and discard less relevant ones during training. In this study, the GRU layer is used to model dependencies along the short feature sequence produced by the convolutional layer. This design enables the capture of structured relationships between input variables with lower architectural complexity than more complex recurrent units. GRUs were selected due to their comparatively lower parameterization and stable training behavior, which are well-suited to the moderate size of the dataset.

The GRU layer has 12 units with `return_sequences = True`, producing an output shape of (5, 12). Several strategies were tested to combine GRU outputs before passing them to the dense layers: full-sequence flattening, which concatenates all hidden states, max pooling, average pooling, and an attention mechanism that computes a weighted combination of the GRU hidden states using learned weights. Specifically, each hidden state  $h_t$  is assigned a weight  $\alpha_t$  calculated as  $\alpha_t = \frac{\exp(w^\top h_t)}{\sum_{i=1}^T \exp(w^\top h_i)}$ , where  $w$  is a trainable parameter vector. The context vector is then computed as  $h_c = \sum_{t=1}^T \alpha_t h_t$ , and is passed to the dense layers of the network to produce the final regression output. The best results were obtained with the full-sequence flattening method, which retains all hidden states without extra parameters. This strategy was particularly effective in our case, likely due to the limited number of input features and the relatively small dataset, allowing it to outperform pooling and attention-based alternatives. The GRU output is flattened into a 60-dimensional vector. This vector is then passed through three fully connected layers with 36, 18, and 9 neurons, using ReLU activation, He-uniform initialization, and L1–L2 regularization parameters set to  $10^{-4}$ . A final sigmoid unit produces the prediction, matching the normalization of compressive strength values to the range [0, 1]. This architecture contains a total of 4827 trainable parameters.

An unsupervised clustering approach was applied to explore the underlying structure of the LWAC dataset and identify subgroups with similar physical and compositional characteristics. To avoid bias from the target variable (compressive strength), clustering was performed exclusively on the seven input variables. Given the mixed data types, the K-prototypes algorithm was employed. The number of clusters was determined using the elbow method by analyzing how cluster compactness evolved as the number of clusters increased, and by identifying the point beyond which further increases resulted in only marginal gains in cluster cohesion. Based on this analysis, three clusters were selected and incorporated as an additional input feature, increasing the total number of parameters to 5259.

### 2.3. Predictive Modeling Process

We used a Monte Carlo cross-validation scheme with 10 random splits, with 75% of the data for training and 25% for testing, to ensure consistent and comparable performance. From each training subset, 5% of the data was reserved for validation.

Conventional k-fold cross-validation is widely adopted and provides performance estimates based on a fixed partition of the dataset. For datasets of moderate size, however, these estimates may depend on the specific partition, indicating that the asymptotic regime has not necessarily been reached [20]. Monte Carlo cross-validation reduces this sensitivity by averaging performance over multiple random splits, leading to more stable estimates as the number of repetitions increases [21]. Repeated k-fold cross-validation can similarly reduce variability but requires multiple model training runs per fold and per repetition, thereby increasing the computational cost. In Monte Carlo cross-validation, the training set size and the number of splits can be specified separately, whereas in k-fold cross-validation, both quantities are coupled by design through the choice of  $k$ . This allows control over the trade-off between stability of the performance estimate and the computational cost, which is relevant when model training is computationally demanding [21].

The features were normalized to the range [0, 1], as described in Section 2.2. The models were trained for up to 800 epochs with early stopping (patience = 50,  $\Delta = 10^{-3}$ ) using mean squared error (MSE) as the loss function. Hyperparameters (optimizer, learning rate, batch size) were tuned based on validation performance across the different splits. The optimal setup used the Adam optimizer ( $\alpha = 10^{-4}$ ,  $\beta_1 = 0.95$ ,  $\beta_2 = 0.999$ ) with a batch size of 32. After restoring the best-performing epoch, the model was retrained on the full training set (75%) for up to 100 epochs, with early stopping applied again ( $\Delta = 10^{-4}$ ). Once training was completed, the predicted outputs were rescaled by applying the inverse of the Min–Max transformation. The performance of the model on the test sets for each Monte Carlo split was then assessed using three standard regression metrics: mean absolute error (MAE), root mean square error (RMSE), and the coefficient of determination ( $R^2$ ).

For a fair comparison, baseline models from the literature were reimplemented and evaluated under identical conditions (same splits, preprocessing, and random seeds). To interpret predictions, we used SHapley Additive exPlanations (SHAP) [22], a method based on cooperative game theory. SHAP values were computed to quantify feature importance, and SHAP visualizations were analyzed to enhance the interpretability and transparency of the model's decision-making process.

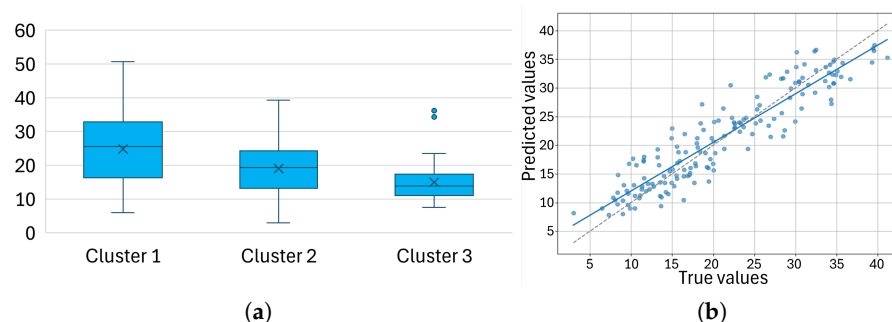
### 3. Results and Discussion

The experimental setup was implemented in Python 3.12 using TensorFlow 2.19 [23]. Additional dependencies included Scikit-learn 1.6 [24], NumPy 1.26, and pandas 2.2. The experiments were conducted on a 64-bit Windows 10 system equipped with an Intel Core i7-1065G7 CPU running at 1.30 GHz, 16 GB of RAM, and an NVIDIA GeForce RTX 3050 GPU with 4 GB of GDDR6 VRAM.

Figure 1a shows the distribution of compressive strength values in the three groups obtained using the K-prototypes algorithm, while Table 4 presents descriptive statistics for all variables within each cluster. As the assumptions of normality were not met, nonparametric statistical tests were applied. The Kruskal–Wallis test revealed significant differences among the groups in terms of compressive strength and several material properties, including LWAC fixed density, LWA particle density, experimental dry density, P-wave velocity, and segregation index ( $p < 0.001$ ). In contrast, process-related variables, such as concrete laying time and vibration time, did not show statistically significant differences among clusters. Pairwise Mann–Whitney U tests with Bonferroni correction confirmed the differences between clusters, except for the LWAC fixed density between clusters 2 and 3.

Given the differences observed between clusters in several key physical variables, two modeling approaches were considered: one using the original seven input variables and another incorporating the cluster label as an additional feature to assess its potential contribution to predictive performance. Table 5 presents the average performance of

the proposed models on the test sets, alongside various machine learning approaches reported in previous studies. The table displays both the original results from the literature and the results of our reimplementations of those models, evaluated under the same experimental conditions as our models. In addition to literature-based baselines, we consider a standalone long short-term memory (LSTM) network as an alternative recurrent architecture, as well as simplified variants of the proposed model, namely a fully connected neural network (FCNN) and a convolutional-dense model (ConvDense), to analyze the contribution of the different architectural components. In all simplified variants, the remaining network components and training settings were kept identical to those of the proposed architecture.



**Figure 1.** Cluster-based distribution of compressive strength and the results of the predictive model. (a) Box plots showing compressive strength across clusters. (b) Predicted versus true compressive strength for the CL-ConvGRU model.

**Table 4.** Summary statistics per variable and cluster.

Variable	Cluster 1	Cluster 2	Cluster 3
Compressive strength	24.84 (9.60), 25.60	19.04 (7.12), 19.34	15.00 (5.47), 13.83
LWAC fixed density	1881.41 (58.17), 1900	1722.30 (63.07), 1700	1724.00 (65.65), 1700
LWA particle density	735.01 (268.48), 482	816.18 (260.81), 1019	482.00 (0.00), 482
Concrete laying time	50.48 (29.49), 60	47.70 (29.77), 30	43.80 (15.99), 30
Vibration time	31.70 (28.53), 20	28.99 (29.78), 20	25.00 (14.74), 20
Experimental dry density	1796.48 (132.48), 1802.94	1544.52 (127.41), 1538.30	1621.27 (147.92), 1637.46
P-wave velocity	3879.59 (153.81), 3861.32	3491.38 (130.48), 3490.50	4749.11 (230.84), 4742.28
Segregation index	1.0045 (0.0339), 1.0025	0.9910 (0.0314), 0.9939	1.0216 (0.0471), 1.0158

Values are presented as mean (standard deviation), median.

**Table 5.** Mean statistical measures on the test data for 10 randomly generated models using a 75/25 split with simple random sampling, compared with established methods reported in the literature.

Model	R <sup>2</sup>	RMSE	MAE
ANN <sup>*b</sup> [5]	0.825	3.745	2.897
MLR <sup>*b</sup> [9]	0.766	4.332	3.396
RT <sup>*b</sup> [9]	0.820	3.808	2.928
KNN <sup>*</sup> [10]	-	4.3365	3.3134
KNN	0.7868	4.1250	3.1751
MLP <sup>*</sup> [10]	0.8045	3.9328	2.9960
MLP	0.8171	3.8117	2.8917
FCNN	0.8218	3.7725	2.9254
RF <sup>*</sup> [10]	0.8135	3.8434	2.9809
RF	0.8210	3.7806	2.9381
GBR <sup>*</sup> [10]	0.8182	3.7959	2.9209
GBR	0.8279	3.7079	2.8962
XGBoost <sup>*</sup> [10]	0.8196	3.7812	2.9020

**Table 5.** *Cont.*

Model	$R^2$	RMSE	MAE
XGBoost	0.8274	3.7131	2.8927
SVR * [10]	0.8124	3.8543	2.9944
SVR	0.8200	3.7890	2.9522
WAE * [10]	0.8220	3.7562	2.8755
WAE	0.8271	3.7109	2.8777
LSTM	0.8312	3.6698	2.8470
ConvDense	0.8327	3.6556	2.8483
ConvGRU	0.8365	3.6385	2.8182
CL-ConvGRU	0.8366	3.6168	2.8171

\* Values from the corresponding references, <sup>b</sup> best model reported in the reference, unmarked values correspond to our reimplementations under the same conditions.

Both models, ConvGRU (without the cluster variable) and CL-ConvGRU (with the cluster variable included as an additional input), outperformed all other methods in terms of  $R^2$ , RMSE, and MAE. On average, incorporating the cluster variable resulted in a slight improvement in RMSE (3.6168) and MAE (2.8171) compared with the ConvGRU model.

To assess the robustness of the selected number of clusters, a sensitivity analysis was conducted by evaluating predictive performance for clustering configurations ranging from two to five clusters (see Table 6). Among the evaluated configurations, the three-cluster model showed the most favorable balance between performance and model simplicity. This result is consistent with the elbow method, which supported the selection of three clusters as an appropriate representation of the underlying data structure.

**Table 6.** Predictive performance on the test data for different clustering configurations.

Clustering Configuration	$R^2$	RMSE	MAE
No clustering (ConvGRU)	0.8365 ± 0.0124	3.6385 ± 0.1814	2.8182 ± 0.1662
2 clusters	0.8355 ± 0.0102	3.6275 ± 0.1426	2.8175 ± 0.1350
3 clusters (CL-ConvGRU)	0.8366 ± 0.0093	3.6168 ± 0.1572	2.8171 ± 0.1455
4 clusters	0.8354 ± 0.0116	3.6285 ± 0.1718	2.8231 ± 0.1492
5 clusters	0.8343 ± 0.0116	3.6399 ± 0.1574	2.8426 ± 0.1448

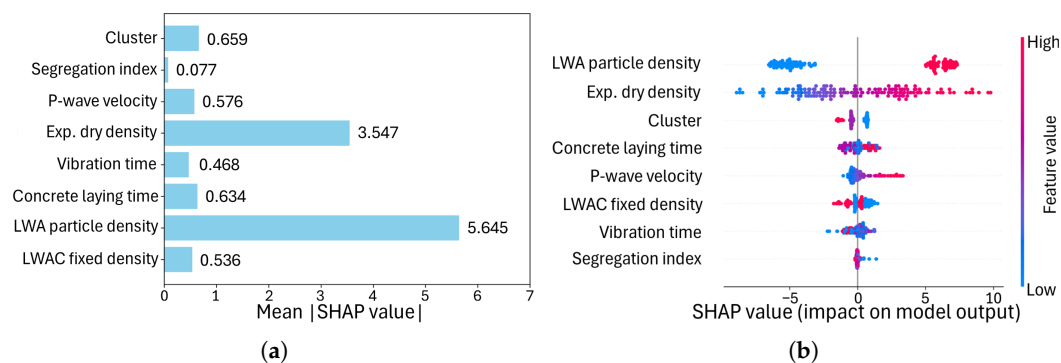
Values are reported as mean ± standard deviation.

Figure 1b shows predicted versus true compressive strength values for the best-performing model across 10 Monte Carlo splits, selected based on the highest  $R^2$  on the test set. For this split, the model achieved an  $R^2$  of 0.8541 and an RMSE of 3.4212.

Figure 2 illustrates the SHAP analysis. The bar plot (Figure 2a) shows the mean absolute SHAP values, ranking features by their average contribution to the model's output. The summary plot (Figure 2b) combines feature importance with the distribution of SHAP values across all test instances, offering a global view of the most influential variables in the model's predictions.

According to the SHAP values, LWA particle density had the greatest impact on model predictions, with higher values generally linked to higher compressive strength predictions. Experimental dry density was the second most influential variable, also showing a positive contribution, indicating that denser mixtures tend to increase predicted strength. The third most important feature was the cluster variable. Cluster 1 increased the predicted compressive strength and was characterized by mixtures with higher measured compressive strength, experimental dry density, and LWAC fixed density, consistent with a denser structure. Cluster 2 caused a moderate decrease in predicted strength, representing an intermediate group with the lowest mean segregation index (0.9910), slightly below 1. Cluster 3 resulted in the largest reduction in predicted strength and corresponded to the

weakest mixtures, characterized by low compressive strength and LWA particle density, high P-wave velocity, and a slightly higher segregation index. Process-related variables such as concrete laying time and vibration time were largely homogeneous across the three clusters, indicating that the cluster variable does not explain differences in these features. SHAP analysis identified concrete laying time as the fourth most influential feature, suggesting its direct contribution to the model's predictions independent of cluster membership. The fifth most important feature was P-wave velocity, whose impact interacted with cluster membership. In cluster 1, SHAP contributions remained positive (up to 0.5) across the entire velocity range, indicating an increase in the predicted compressive strength. In cluster 2, where P-wave velocity values were below 3881 m/s, being in this group moderately reduced the predicted compressive strength (SHAP between  $-0.6$  and  $-0.3$ ). In cluster 3, where values exceeded 4375 m/s, the reduction in predicted compressive strength was stronger (SHAP below  $-1$ ); that is, P-wave velocities in Cluster 2 moderately reduced the predicted compressive strength, while higher velocities in Cluster 3 led to a stronger decrease.



**Figure 2.** SHAP analysis. (a) Bar plot of mean absolute SHAP values. (b) SHAP summary plot.

From an interpretability perspective, it is also important to consider the implications of incorporating cluster information into the model. Because the cluster membership variable is derived from the original input features, its inclusion may introduce redundancy rather than multicollinearity in the classical linear modeling sense. However, empirical results suggest that its contribution is complementary rather than dominant. Specifically, the SHAP analysis indicates that the cluster variable does not replace the main physical predictors but instead modulates their effects within specific ranges. Moreover, the performance gains associated with its inclusion remain moderate. Finally, the use of L1–L2 regularization in the dense layers helps mitigate potential redundancy effects.

#### 4. Conclusions

Hybrid deep learning architectures that combine convolutional, GRU, and dense layers were evaluated for predicting the compressive strength of LWAC. Both ConvGRU and CL-ConvGRU outperformed traditional machine learning and ensemble methods. Incorporating the cluster-derived feature into CL-ConvGRU allowed the detection of latent patterns, enhanced interpretability through SHAP analysis, and highlighted characteristics relevant to LWAC behavior.

Future work will explore the integration of tabular and clustering data with imaging information to enhance predictions and uncover microstructural patterns. Applying this approach to other LWAC properties, aggregate types, and potentially other concrete systems could broaden its practical impact in engineering practice.

**Author Contributions:** Conceptualization, V.M., H.P. and J.P.; methodology, V.M., H.P. and J.P.; software, V.M., H.P. and J.P.; validation, V.M., H.P. and J.P.; formal analysis, V.M., H.P. and J.P.; investigation, V.M., H.P. and J.P.; data curation, V.M., H.P. and J.P.; writing—original draft preparation, V.M., H.P. and J.P.; writing—review and editing, V.M., H.P. and J.P.; supervision, V.M. and J.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is part of the R+D+i projects PID2021-123627OB-C55 and PID2024-158682OB-C33, funded by MCIN/AEI/10.13039/501100011033, “ERDF A way of making Europe”, and “ERDF/EU”.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used in this study were provided by the Materials and Territory Technology research group at the University of Alicante and remain the property of their original owners.

**Acknowledgments:** We thank Antonio J. Tenza-Abril of the Materials and Territory Technology research group at the University of Alicante, for providing the data used in this study.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Chandra, S.; Berntsson, L. *Lightweight Aggregate Concrete*; Elsevier: Amsterdam, The Netherlands, 2002.
2. Tenza-Abril, A.; Benavente, D.; Pla, C.; Baeza-Brotons, F.; Valdes-Abellan, J.; Solak, A. Statistical and experimental study for determining the influence of the segregation phenomenon on physical and mechanical properties of lightweight concrete. *Constr. Build. Mater.* **2020**, *238*, 117642. [[CrossRef](#)]
3. Li, Z.; Yoon, J.; Zhang, R.; Rajabipour, F.; Srubar, W.V., III; Dabo, I.; Radlińska, A. Machine learning in concrete science: Applications, challenges, and best practices. *NPJ Comput. Mater.* **2022**, *8*, 127. [[CrossRef](#)]
4. Ni, B.; Rahman, M.Z.; Guo, S.; Zhu, D. A review on properties and multi-objective performance predictions of concrete based on machine learning models. *Mater. Today Commun.* **2025**, *44*, 112017. [[CrossRef](#)]
5. Tenza-Abril, A.J.; Villacampa, Y.; Solak, A.M.; Baeza-Brotons, F. Prediction and sensitivity analysis of compressive strength in segregated lightweight concrete based on artificial neural network using ultrasonic pulse velocity. *Constr. Build. Mater.* **2018**, *189*, 1173–1183. [[CrossRef](#)]
6. Hemmatian, A.; Jalali, M.; Naderpour, H.; Nehdi, M.L. Machine learning prediction of fiber pull-out and bond-slip in fiber-reinforced cementitious composites. *J. Build. Eng.* **2023**, *63*, 105474. [[CrossRef](#)]
7. Chen, Z.; Zhang, L.; Li, K.; Xue, X.; Zhang, X.; Kim, B.; Li, C.Y. Machine-learning prediction of aerodynamic damping for buildings and structures undergoing flow-induced vibrations. *J. Build. Eng.* **2023**, *63*, 105374. [[CrossRef](#)]
8. Sajjan, K.C.; Bhusal, A.; Gautam, D.; Rupakhety, R. Earthquake damage and rehabilitation intervention prediction using machine learning. *Eng. Fail. Anal.* **2023**, *144*, 106949. [[CrossRef](#)]
9. Migallón, V.; Navarro-González, F.J.; Penadés, J.; Villacampa, Y. Parallel approach of a Galerkin-based methodology for predicting the compressive strength of the lightweight aggregate concrete. *Constr. Build. Mater.* **2019**, *219*, 56–68. [[CrossRef](#)]
10. Migallón, V.; Penadés, H.; Penadés, J.; Tenza-Abril, A.J. A machine learning approach to prediction of the compressive strength of segregated lightweight aggregate concretes using ultrasonic pulse velocity. *Appl. Sci.* **2023**, *13*, 1953. [[CrossRef](#)]
11. Kewalramani, M.A.; Gupta, R. Concrete compressive strength prediction using ultrasonic pulse velocity through artificial neural networks. *Autom. Constr.* **2006**, *15*, 374–379. [[CrossRef](#)]
12. Tavakkol, S.; Alapour, F.; Kazemian, A.; Hasaninejad, A.; Ghanbari, A.; Ramezaniyanpour, A.A. Prediction of lightweight concrete strength by categorized regression, MLR and ANN. *Comput. Concr.* **2013**, *12*, 151–167. [[CrossRef](#)]
13. Charhate, S.; Subhedar, M.; Adsul, N. Prediction of Concrete Properties Using Multiple Linear Regression and Artificial Neural Network. *J. Soft Comput. Civ. Eng.* **2018**, *2*, 27–38. [[CrossRef](#)]
14. Deshpande, N.; Londhe, S.; Kulkarni, S. Modeling compressive strength of recycled aggregate concrete by Artificial Neural Network, Model Tree and Non-linear Regression. *Int. J. Sustain. Built Environ.* **2014**, *3*, 187–198. [[CrossRef](#)]
15. Migallón, V.; Penadés, H.; Penadés, J. Key Predictors of Lightweight Aggregate Concrete Compressive Strength by Machine Learning from Density Parameters and Ultrasonic Pulse Velocity Testing. *Mater. Proc.* **2025**, *26*, 4. [[CrossRef](#)]

16. Guzmán-Torres, J.A.; Domínguez-Mota, F.J.; Tinoco-Guerrero, G.; Tinoco-Ruíz, J.G.; Alonso-Guzmán, E.M. Extreme fine-tuning and explainable AI model for non-destructive prediction of concrete compressive strength, the case of ConcreteXAI dataset. *Adv. Eng. Softw.* **2024**, *192*, 103630. [[CrossRef](#)]
17. Gan, X.; Wang, W.; Jiang, C.; Ye, L.; Chen, F.; Zhou, T.; Zhao, Y. Ultrasonic detection and deep learning for high-precision concrete strength prediction. *J. Build. Eng.* **2025**, *104*, 112372. [[CrossRef](#)]
18. Fernández-Fanjul, A.; Tenza-Abril, A.J. Méthode Fanjul: Dosage pondéral des bétons légers et lourds. *Ann. Bâtim. Trav. Publics* **2012**, *5*, 32–50.
19. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*; Wu, D., Carpuat, M., Carreras, X., Vecchi, E.M., Eds.; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 103–111. [[CrossRef](#)]
20. Xu, Y.; Goodacre, R. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *J. Anal. Test.* **2018**, *2*, 249–262. [[CrossRef](#)] [[PubMed](#)]
21. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [[CrossRef](#)]
22. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 4765–4774.
23. Ünsalan, C.; Höke, B.; Atmaca, E. The TensorFlow Platform and Keras API. In *Embedded Machine Learning with Microcontrollers: Applications on STM32 Development Boards*; Springer International Publishing: Cham, Switzerland, 2025; pp. 215–228. [[CrossRef](#)]
24. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.