Physics-Informed Loss Functions for Enhancing Concrete Compressive Strength Prediction with Neural Networks

Oğuz Akif Tüfekcioğlu
Dept. of Computer Engineering
Hacettepe University
Ankara, Turkey
oguztufekcioglu@hacettepe.edu.tr

Mehmet Önder Efe
Dept. of Computer Engineering
Hacettepe University
Ankara, Turkey
onderefe@hacettepe.edu.tr

***Abstract*—This study investigates the application of physics informed loss functions in the prediction of concrete compressive strength using neural networks. By integrating domain knowledge about monotonic relationships and other constraints into the loss calculation, we demonstrate improved model performance under data-scarce conditions. Comparisons are drawn against baseline neural networks to highlight the effectiveness of the physics-informed approach.**

***Index Terms*—Neural Networks, Physics-Informed Neural Networks, Loss Optimization, Data Scarcity**

I. Introduction

Predicting the compressive strength of concrete is a key challenge in civil engineering, as it directly impacts the safety and efficiency of construction projects. The relationship between the material properties of concrete, such as water-cement ratio, cement content, and other additives, and its compressive strength is governed by well-established physical laws. While traditional machine learning models, including neural networks, have been successfully applied to this problem, they often treat the task as purely data-driven, neglecting the underlying physics. This limitation can lead to overfitting and poor generalization, particularly in cases where training data is scarce.

To address these challenges, we propose incorporating domain-specific physical laws into the training process of neural networks through a physics-informed loss function. By embedding constraints such as the monotonicity of the water-cement ratio (i.e., higher ratios generally lead to lower compressive strength) and the inverse relationship between cement content and strength, our approach ensures that the model adheres to established scientific principles while still leveraging data-driven learning.

The physics-informed loss function modifies the training objective by penalizing violations of these physical laws, guiding the network to learn patterns that are both accurate and physically plausible. This approach not only improves prediction accuracy but also reduces the reliance on large training datasets, as the physical laws compensate for limited data availability.

In this work, we evaluate the proposed method on the UCI Concrete Compressive Strength dataset, a widely used benchmark for this task. Through comparative experiments, we demonstrate that the physics-informed neural network outperforms standard networks in terms of validation loss, even when trained on reduced datasets. By aligning the model's learning process with physical realities, this method contributes to more robust, interpretable, and reliable predictions, offering significant advantages for real-world applications in material science and engineering.

o deterministic formulas while still respecting governing constraints through the proposed loss.

II. Related Work

The task of predicting the compressive strength of concrete has attracted considerable attention, with various machine learning and neural network-based approaches applied to this problem. This section reviews significant studies and highlights the contributions of our proposed method.

A. Neural Network-Based Models

Artificial Neural Networks (ANNs) have been widely used for modeling the relationship between material properties and concrete compressive strength. Asteris and Mokos (2020) proposed an ANN-based approach to predict compressive strength, demonstrating its effectiveness in capturing non-linear relationships in the data. Their study highlighted the challenges of generalization in ANN models due to their reliance on large amounts of training data [1]. Similarly, Varma et al. (2023) explored deep learning models for compressive strength prediction, emphasizing feature selection and hyper-parameter tuning to improve accuracy [2].

In addition, Motlagh and Naghizadehrokni [3] addresses a critical issue in neural network training—the availability and quality of data. This study shows that while ANNs and other machine learning models can effectively learn the complex relationships in concrete mixtures, their performance is significantly influenced by the size and consistency of the available datasets. The findings suggest that sparse data can limit the predictive capability of neural network-based models, thereby motivating the need for either data augmentation or the integration of hybrid approaches.

B. Deep Learning with Advanced Architectures

Deep learning methods, such as Convolutional Neural Networks (CNNs), have also been applied to this domain. Naved et al. (2024) utilized CNNs to model the compressive strength of concrete, leveraging their capability to extract meaningful patterns from raw data. While CNNs achieved high accuracy, their high computational requirements and lacking of physical interpretability remain key limitations [4]. Additionally, Latif (2021) applied Long Short-Term Memory Networks (LSTMs) to model time-dependent properties of concrete strength. These models excel in capturing temporal dependencies but do not account for domain-specific physical laws [5].

C. Hybrid and Optimized Models

Metaheuristic optimization techniques have been integrated into neural network training to enhance prediction performance. Shishegaran et al. (2021) proposed a hybrid model combining ANNs with metaheuristic optimization, such as adaptive neuro-fuzzy inference systems (ANFIS), to address challenges in predicting concrete properties. Their work demonstrated improved accuracy, but introduced increased computational complexity [6].

Another study, [7] focuses on a specialized application involving confined concrete. By employing various metaheuristic algorithms (including particle swarm optimization, grey wolf optimizer, and bat algorithm) alongside an ANN, this study achieves very high predictive accuracy for the confinement effect imparted by CFRP systems. The success of these hybrid models underscores the potential of combining metaheuristic optimization with neural networks to address both data limitations and model generalization issues.

D. Physics-Informed Neural Networks (PINNs)

Physics-Informed Neural Networks (PINNs) represent a significant advancement in machine learning by integrating domain-specific knowledge, such as physical laws or governing equations, into the training process. Raissi et al. (2019) introduced PINNs as a framework to solve forward and inverse problems involving partial differential equations (PDEs), demonstrating their ability to improve generalization by respecting underlying physics [8]. By embedding physical constraints into the loss function, PINNs enable models to generate predictions that are not only accurate but also physically plausible.

Applications of PINNs have been widely explored in diverse domains:

- **Fluid Dynamics:** PINNs have been used to solve the Navier-Stokes equations to model fluid flow, achieving high accuracy while requiring less data compared to traditional data-driven models [9].
- **Structural Analysis:** Lu et al. (2021) applied PINN to the prediction of stress-strain in materials, where physical laws governed by the theory of elasticity were embedded in the training [10].
- **Climate and Weather Modeling:** PINNs have also been utilized to model the dynamics of temperature and precipitation, incorporating thermodynamic constraints to improve reliability [11].

Recently, several groups have tailored PINNs to cementitious materials. Varghese et al. [12] optimised mix cost while satisfying hydration physics. Rahman & Lu [13] introduced PINN-CHK for early-age hydration kinetics. Ke et al. [14] simulated mechanical responses under load using a PINN formulation OUCI. Mobasheri et al. [15] offered a comprehensive 2025 survey of ML control strategies for concrete performance. Compared with these works we target compressive-strength regression rather than process optimisation or hydration temperature, and show that very small penalty terms already bring pronounced gains in standard feed-forward nets, avoiding the complexity of PDE residual networks.

E. Applications in Material Science and Engineering

Within material science, the use of physics-informed approaches has shown significant promise. Haghighat et al. (2021) demonstrated the application of PINNs in modeling heterogeneous material properties by embedding stress-strain relationships directly into the network [16]. This approach not only improved accuracy but also provided physically interpretable results.

For concrete compressive strength prediction, physics-informed methods are particularly valuable due to the well-established relationships between material properties and strength. While traditional machine learning models focus purely on data-driven patterns, incorporating constraints such as monotonicity or inverse relationships aligns predictions with physical laws, improving robustness in data-scarce scenarios.

F. Contribution of This Work

While prior studies primarily focus on purely data-driven or hybrid models, they do not explicitly incorporate domain-specific physical principles. This gap is addressed in our study by introducing a physics-informed loss function that embeds physical constraints, such as monotonicity of the water-cement ratio and the inverse relationship between cement content and compressive strength, into the neural network training process. By penalizing violations of these constraints, our approach ensures predictions remain physically plausible, improves validation loss, and reduces the dependency on large datasets. This novel integration of domain knowledge sets our work apart from existing methodologies.

III. Data Preprocessing

The UCI Concrete Compressive Strength dataset contains nine columns: eight input features (cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, and age) and one target column (compressive strength). Effective data preprocessing is critical to ensure meaningful results during model training and evaluation.

A. Feature Scaling

Since the input features vary significantly in scale (e.g., cement and superplasticizer), standardization was applied to bring all features to a mean of 0 and a standard deviation of 1. This was achieved using a custom-built StandardScaler module, which computed the mean and standard deviation for each feature and applied the transformation as follows:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma} \tag{1}$$

where,

- X: Original feature value
- μ: Mean of the feature values
- σ: Standard deviation of the feature values

B. Target Scaling

The target variable (compressive strength) was also standardized to improve convergence during training. This scaling was reversed during evaluation to report results in the original units.

C. Train-Test Splitting

The dataset was split into training and testing sets using an 80%, 20% ratio. To ensure reproducibility, the data was shuffled and split with a fixed random seed. The training set was further divided to create a smaller subset (90% of the training data) for experiments involving reduced data, simulating scenarios with limited training samples.

IV. Neural Network Architectures and Physics Penalty Functions

This section describes the neural network architectures explored and the physics-informed penalty functions integrated into the loss computation during training. These configurations were designed to assess the impact of embedding physical constraints into the learning process for predicting concrete compressive strength.

A. Neural Network Architectures

Three distinct architectures were evaluated, differing in the number of layers and neurons, to capture varying levels of model complexity:

- **Architecture 1 (Arch1):**
  - Input Layer: 8 features (All input features from UCI Concrete Dataset)
  - Hidden Layer 1: 64 neurons, ReLU activation
  - Output Layer: 1 neuron (compressive strength prediction)
- **Architecture 2 (Arch2):**
  - Input Layer: 8 features (All input features from UCI Concrete Dataset)
  - Hidden Layer 1: 128 neurons, ReLU activation
  - Hidden Layer 2: 64 neurons, ReLU activation
  - Output Layer: 1 neuron
- **Architecture 3 (Arch3):**
  - Input Layer: 8 features (All input features from UCI Concrete Dataset)
  - Hidden Layer 1: 128 neurons, ReLU activation
  - Hidden Layer 2: 128 neurons, ReLU activation
  - Output Layer: 1 neuron

These architectures progressively increase in complexity, with Arch3 being the most expensive model. The output layer in all cases predicts a single value representing the compressive strength of concrete.

B. Physics-Informed Penalty Functions

To incorporate domain-specific physical constraints into the learning process, penalty functions were added to the loss function. These penalties enforce relationships between features and the target variable based on known physical principles:

- **Monotonicity with Water-Cement Ratio (Global):**
  Enforces that higher water-cement (W/C) ratios do not produce higher compressive strength. A global monotonicity penalty ensures predictions decrease or remain constant with increasing W/C ratios across the dataset.

$$P_{WC} = \frac{1}{N-1} \sum_{i=1}^{N-1} \max\Big(y_{i+1}^{(WC)} - y_i^{(WC)},\, 0\Big)^2 \tag{2}$$

  where,
  - $y_i^{(WC)}$: the predicted compressive strength, sorted by increasing W/C ratio.
  - N: Number of samples in the dataset.
- **Inverse Relation with Cement Content:**
  Penalizes violations of the inverse relationship between cement content and compressive strength. Higher cement content should not lead to a decrease in predicted strength.

$$P_{C} = \frac{1}{N-1} \sum_{i=1}^{N-1} \max\Big(y_{i+1}^{(C)} - y_i^{(C)},\, 0\Big)^2 \tag{3}$$

  where,
  - $y_i^{(C)}$: Predicted compressive strength, sorted by decreasing cement content.
  - N: Number of samples in the dataset.
- **Monotonicity with W/C Ratio (Local):**
  Enforces monotonicity by comparing random pairs of samples in a batch. If a sample with a higher W/C ratio predicts a higher strength than one with a lower ratio, a penalty is applied.

The penalty functions are scaled by a hyperparameter $\lambda_{\text{pi}} > 0$ to balance their influence on the overall loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{pi}} P$$

C. Integration into Training

During training, these penalties were computed dynamically for each batch and added to the Mean Squared Error (MSE) loss. The penalty gradients were propagated alongside the

MSE gradients, allowing the network to learn predictions that adhered to physical laws.

This setup enabled a fair comparison between the baseline (purely data-driven) models and the physics-informed models, demonstrating the value of embedding domain knowledge into the training process.

D. Hyperparameter Optimization

To identify the optimal training configuration for each neural network architecture, hyperparameter optimization was performed for 1000 epoches. The process involved exploring a range of learning rates (lr), momentum values, and batch sizes. The best combination for each architecture was selected based on the lowest validation loss. The hyperparameter search space included:

- **Learning Rates** (lr): $\{0.001, 0.0005, 0.0001\}$
- **Momentum Values**: $\{0.8, 0.85, 0.9, 0.95\}$
- **Batch Sizes**: $\{16, 32, 64\}$

The results of the hyperparameter optimization for each architecture are presented in Table I. The configurations achieving the lowest validation loss for each architecture were used for training the final models.

TABLE I
HYPERPARAMETER OPTIMIZATION RESULTS FOR NEURAL NETWORK ARCHITECTURES

Architecture	lr	Momentum	Batch Size	Val Loss
Arch1	0.001	0.95	16	0.0637
Arch2	0.001	0.95	16	0.0582
Arch3	0.001	0.95	16	0.0550

Optimization Insights:

- Among all architectures, the best performance was observed with a learning rate of 0.001, momentum of 0.95, and a batch size of 16.
- As the architecture complexity increased (from Arch1 to Arch3), the validation loss consistently decreased, indicating that more complex architectures were better suited to capture the underlying patterns in the data.

These optimized hyperparameters were employed to train both the baseline and physics-informed models, ensuring fair and effective comparisons.

V. EXPERIMENTS

This section presents the results of the experimental evaluation of three neural network architectures (Arch1, Arch2, and Arch3) under both baseline and physics-informed training regimes. Each architecture was trained with two physics constraints: the inverse relationship with cement content and the monotonicity of the Water-Cement ratio. The experiments aim to evaluate the effectiveness of physics-informed loss functions in improving validation performance and generalization.

A. Training Setup

- **Architectures:** Three progressively complex architectures were evaluated:
  - **Arch1:** A single hidden layer with 64 neurons.
  - **Arch2:** Two hidden layers with 128 and 64 neurons, respectively.
  - **Arch3:** Two hidden layers with 128 neurons each.
- **Physics-Informed Constraints:**
  - **Inverse Cement Relation:** Enforces that higher cement content does not result in lower compressive strength.
  - **Monotonic W/C Ratio:** Ensures that a higher water-cement ratio does not produce higher compressive strength.
- **Metrics:** Validation loss (MSE) and validation loss difference (baseline vs. physics-informed) were used to compare performance across models and constraints.

VI. RESULTS

A. Validation Loss Trends

The validation loss trends highlight the performance improvements of physics-informed models. Fig. 1 shows the validation loss over epochs for the best-performing architecture (Arch2) with the Inverse Cement constraint. The physics-informed model demonstrates consistently lower validation loss compared to the baseline, particularly in later epochs.

This improvement can be attributed to the integration of domain-specific constraints into the learning process. By embedding the inverse relationship between cement content and compressive strength as a penalty in the loss function, the physics-informed model aligns its predictions with established physical principles. This ensures that the model does not overfit to spurious patterns in the training data, which is a common challenge in data-driven methods, especially under data scarcity.

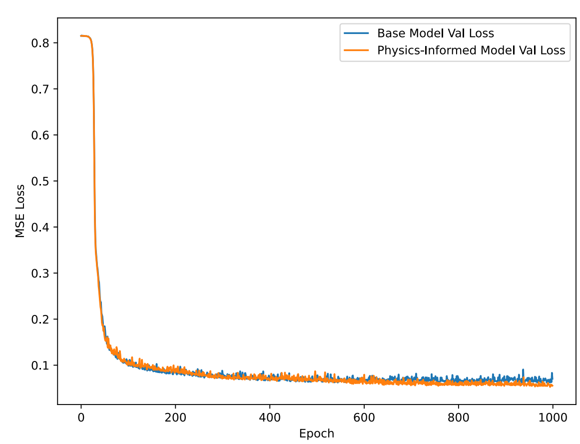

Fig. 1. Validation loss over epochs for Arch2 with Inverse Cement constraint.

B. Validation Loss Differences

The validation loss differences between baseline and physics-informed models, under both reduced and full training data conditions, are summarized in Tables II and III. These results demonstrate the robustness of physics-informed constraints in improving model performance, even with limited data.

In Tables II and III, two distinct metrics are reported to evaluate the validation loss at different stages of training:

- Early Mean: This is the average validation loss computed over the initial training phase, specifically from epoch 0 to 200. It reflects the model's performance at the beginning of the training process and shows how well the model performs before it has fully converged.
- Final Mean: This is the average validation loss computed over the final training phase, specifically from epoch 800 to 1000. It indicates the model's performance after extensive training and convergence, demonstrating the impact of the physics-informed constraints on the later stages of learning.

Comparing these two metrics helps assess how the model's behavior and the effect of the constraints evolve from the early to the late training phases.

TABLE II
VALIDATION LOSS DIFFERENCES WITH 90% TRAINING DATA

Architecture	Constraint	Early Mean	Final Mean	Overall Mean ± Std
Arch1	Inverse_cement	0.0010	-0.0052	-0.0015 ± 0.0063
Arch1	Monotonic_w_c	-0.0007	-0.0104	-0.0031 ± 0.0078
Arch2	Inverse_cement	-0.0075	0.0007	-0.0040 ± 0.0093
Arch2	Monotonic_w_c	-0.0012	0.0067	0.0011 ± 0.0080
Arch3	Inverse_cement	-0.0067	0.0026	0.0013 ± 0.0118
Arch3	Monotonic_w_c	-0.0107	-0.0009	-0.0034 ± 0.0182

TABLE III
VALIDATION LOSS DIFFERENCES WITH FULL TRAINING DATA

Architecture	Constraint	Early Mean	Final Mean	Overall Mean ± Std
Arch1	Inverse_cement	0.0031	-0.0012	0.0001 ± 0.0062
Arch1	Monotonic_w_c	-0.0012	-0.0187	-0.0107 ± 0.0090
Arch2	Inverse_cement	-0.0009	0.0084	0.0029 ± 0.0071
Arch2	Monotonic_w_c	0.0063	0.0063	0.0044 ± 0.0094
Arch3	Inverse_cement	0.0030	0.0044	0.0039 ± 0.0051
Arch3	Monotonic_w_c	0.0087	-0.0056	0.0005 ± 0.0083

C. Final MSE Comparison Across Architectures

To provide a clear comparison of the final MSE losses for each experiment, Figs. 2 and 3 present the results grouped by architecture. These visualizations highlight the performance differences between baseline and physics-informed models under both full and reduced training data conditions.

Architecture Findings:

- **Impact of Constraints:** Across all architectures, the physics-informed models consistently outperform their baseline counterparts, with the Inverse Cement constraint yielding the most significant improvements.
- **Architecture Complexity:** The performance gains from physics-informed constraints are more pronounced in simpler architectures like Arch1, compensating for their limited capacity. However, deeper architectures like Arch3 achieve the lowest overall MSE when paired with sufficient training data.
- **Optimal Configuration:** Arch2 with the Inverse Cement constraint strikes the best balance between complexity and generalization, emerging as the most robust architecture under both full and reduced data scenarios.
- **Data Scarcity Resilience:** Despite using only 90% of the training data, the physics-informed models demonstrate competitive performance, with smaller performance drops compared to baseline models.

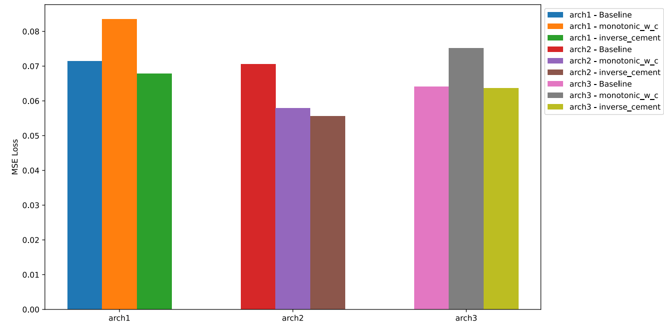

Fig. 2. Final MSE loss for baseline and physics-informed models (Full Training Data).

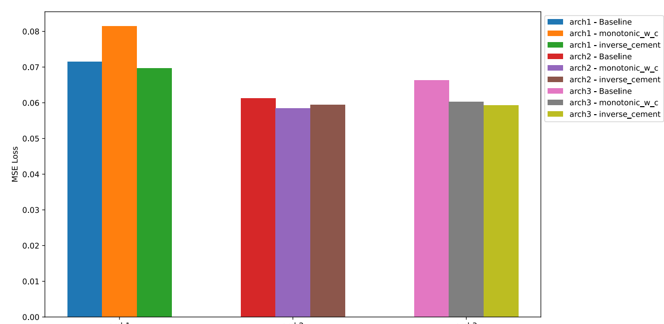

Fig. 3. Final MSE loss for baseline and physics-informed models (90% Training Data).

D. Key Insights

- The Arch2 model with the Inverse Cement constraint achieved the best performance across all training scenarios, demonstrating the value of incorporating physical constraints into neural networks.
- Physics-informed loss functions consistently improved validation loss, particularly during later epochs, across all architectures and training data conditions.
- Under data scarcity conditions (90% training data), the Monotonic W/C Ratio constraint showed significant improvements in simpler architectures, compensating for reduced model capacity.
- Deeper architectures like Arch2 benefited most from physics-informed loss, achieving the lowest overall validation loss in experiments with full training data.

VII. Discussion

The results demonstrate the advantages of physics-informed loss functions in improving the generalization and robustness of neural networks for predicting concrete compressive strength. By embedding domain-specific physical constraints into the training process, the models consistently achieved better validation performance than their baseline counterparts.

Significance of the Inverse Cement Constraint: The Inverse Cement constraint effectively aligned model predictions with established physical relationships, leading to superior performance in both data-rich and data-scarce scenarios. This was particularly evident in the Arch2 model, which balanced architectural complexity with effective generalization.

Role of Architecture Complexity: Role of Architecture Complexity: While deeper architectures like Arch3 achieved the best overall performance with full training data, simpler architectures such as Arch1 benefited most from physics-informed constraints under data scarce conditions. This demonstrates the flexibility of physics informed loss functions to adapt to varying levels of model complexity. Future studies could explore extending this approach to other domains with well-defined physical relationships or integrating additional domain knowledge to improve model reliability further.

VIII. Conclusion

This study investigated the integration of domain-specific physical knowledge into neural network training through physics-informed loss functions for predicting concrete compressive strength. By embedding constraints such as the Inverse Cement relation and Monotonic Water-Cement Ratio, the proposed approach demonstrated significant improvements in model performance and generalization, particularly in data-scarce scenarios.

Important findings include:

- The Arch2 model with the Inverse Cement constraint emerged as the best-performing configuration, achieving the lowest validation loss across all experiments and demonstrating the efficacy of aligning model training with physical relationships in the data.
- Physics-informed models consistently outperformed their baseline counterparts, particularly during the later epochs, highlighting their ability to generalize better while adhering to established physical principles.
- Under data scarcity conditions (90% training data), physics-informed loss functions mitigated performance degradation, with simpler architectures benefiting most from the additional constraints.

These results underline the potential of physics-informed neural networks to enhance predictive accuracy and reliability while reducing dependency on large datasets. The integration of physical constraints not only improves model performance but also ensures predictions align with domain-specific knowledge, fostering greater trust and interpretability.

Future work could focus on extending this framework to other engineering and scientific domains where well-defined physical laws govern system behavior. Additionally, exploring ensemble methods or integrating other machine learning paradigms, such as unsupervised learning or transfer learning, may further enhance the applicability and scalability of physics-informed loss functions.

References

[1] P. G. Asteris and V. G. Mokos, "Concrete compressive strength using artificial neural networks," *Neural Computing and Applications*, vol. 32, no. 9, pp. 11 807–11 826, 2020.

[2] B. V. Varma, E. V. Prasad, and S. Singha, "Study on predicting compressive strength of concrete using supervised machine learning techniques," *Asian Journal of Civil Engineering*, vol. 24, no. 7, pp. 2549–2560, 2023.

[3] B. Ouyang, Y. Li, Y. Song, F. Wu, H. Yu, Y. Wang, M. Bauchy, and G. Sant, "Learning from sparse datasets: Predicting concrete's strength by machine learning," *CoRR*, vol. abs/2004.14407, 2020. [Online]. Available: https://arxiv.org/abs/2004.14407

[4] M. Naved, M. Asim, and T. Ahmad, "Prediction of concrete compressive strength using deep neural networks based on hyperparameter optimization," *Cogent Engineering*, vol. 11, no. 1, 2024.

[5] S. D. Latif, "Concrete compressive strength prediction modeling utilizing deep learning long short-term memory algorithm for a sustainable environment," *Environmental Science and Pollution Research*, vol. 28, no. 7, pp. 30 294–30 302, 2021.

[6] A. Shishegaran, H. Varaee, T. Rabczuk, and G. Shishegaran, "High correlated variables creator machine: Prediction of the compressive strength of concrete," *Computers & Structures*, vol. 247, p. 106479, 2021.

[7] S. Wahab, M. Suleiman, F. Shabbir, N. S. Mahmoudabadi, S. Waqas, N. Herl, and A. Ahmad, "Predicting confinement effect of carbon fiber reinforced polymers on strength of concrete using metaheuristics-based artificial neural networks," 2023. [Online]. Available: https://arxiv.org/abs/2403.13809

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving pdes," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[9] M. Raissi, "Deep hidden physics models: Deep learning of nonlinear partial differential equations," *Journal of Machine Learning Research*, 2018.

[10] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and G. E. Karniadakis, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113741, 2021.

[11] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *SIAM Review*, 2021.

[12] S. Varghese, R. Anand, and G. Paliwal, "Physics-informed neural network for concrete manufacturing process optimization," *arXiv preprint arXiv:2408.14502*, 2024.

[13] M. A. Rahman, T. Zhang, and Y. Lu, "Pinn-chk: Physics-informed neural network for high-fidelity prediction of early-age cement hydration kinetics," *Neural Computing and Applications*, vol. 36, no. 22, pp. 13 665–13 687, 2024.

[14] Y. Ke, C. Liu, and S.-S. Zhang, "Physics-informed neural network (pinn)-based numerical simulation of concrete mechanical responses," *Available at SSRN 5162809*.

[15] F. Mobasheri, M. Hosseinpoor, A. Yahia, and F. Pourkamali-Anaraki, "Machine learning as an innovative engineering tool for controlling concrete performance: A comprehensive review," *Archives of Computational Methods in Engineering*, pp. 1–45, 2025.

[16] T. Beucler, M. Pritchard, and P. Gentine, "Enforcing analytic constraints in neural networks emulating physical systems," *Physical Review Letters*, vol. 126, no. 9, p. 098302, 2021.