BILINGUAL PUBLISHING GROUP

Journal of Building Material Science

https://journals.bilpubgroup.com/index.php/jbms

ARTICLE

Comparative Analysis of Prediction Model for Non destructive Testing based Compressive Strength Determination

Priyesh Gangele , ***Arun Kumar Patel*** *

Department of Civil Engineering, Vedica Institute of Technology, RKDF University, Bhopal, Madhya Pradesh 462033, India

ABSTRACT

Evaluating the performance of existing concrete structures is essential in civil engineering, with compressive strength serving as an indicator of performance. Non-destructive testing (NDT) techniques are commonly employed due to their cost-effectiveness and the ability to assess structural integrity without causing damage. However, NDT methods often yield less accurate results than destructive testing (DT), which, although highly reliable, is costly and invasive. To address this limitation, recent research has focused on developing predictive models that correlate DT and NDT outcomes using machine learning techniques. This study explores the application of Support Vector Machine (SVM) models, enhanced with optimization techniques, to improve prediction accuracy. Experimental concrete practical samples, ranging from M10 to M40 grade, were prepared and tested at 14 and 28 days of curing, totaling 126 laboratory specimens. Additionally, 231 field samples were collected from a 20-year-old structure to reflect in situ conditions. The performance of SVM was improved using optimization algorithms such as Bayesian Optimization and Genetic Algorithms (GA). Among various kernel functions tested, the Gaussian non-linear kernel proved most effective in modeling the complex relationship between NDT and DT results. The SVM model optimized using Bayesian methods and a Gaussian kernel achieved superior performance, with a high coefficient of determination ($R^2 = 0.9771$) and significantly lower error metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Bayesian-optimized SVM

*CORRESPONDING AUTHOR:

Arun Kumar Patel, Department of Civil Engineering, Vedica Institute of Technology, RKDF University, Bhopal, Madhya Pradesh 462033, India ; Email: arunpatel123@gmail.com

ARTICLE INFO

Received: 19 May 2025 | Revised: 18 June 2025 | Accepted: 25 June 2025 | Published Online: 24 July 2025
DOI: https://doi.org/10.30564/jbms.v7i3.10097

CITATION

Gangele, P., Patel, A.K., 2025. Comparative analysis of prediction model for non destructive testing based compressive strength determination. Journal of Building Material Science. 7(3): 62–80. DOI: https://doi.org/10.30564/jbms.v7i3.10097

with a Gaussian kernel offers a highly accurate and practical tool for predicting compressive strength from NDT data, enhancing decision-making in structural assessment.

Keywords: Destructive Test; Non-Destructive Testing; Support Vector Machine (SVM); Bayesian-Optimized SVM; Genetic Algorithm-Optimized SVM; ANN

1. Introduction

The repair and refurbishment of existing building structures have become essential research fields in current engineering construction due to the growing expansion of high-rise buildings and infrastructure construction on a national and international scale [1]. Concrete, one of the many building materials used in construction, uses its compressive strength as the main indicator of structural integrity [2,3]. Destructive testing (DT) is typically the primary approach used to measure the compressive strength of concrete [4–7]. DT, however, is less practical for decaying concrete structures since they not only come with a significant cost but also jeopardize the functionality of the current structure [7,8]. To evaluate the quality of concrete structures, particularly in terms of compressive strength, non-destructive testing (NDT) techniques have become increasingly widespread [9–11]. When compared to DT, NDT yields less accurate and dependable results while lowering testing expenses, workload, and harm to the original structure [12]. Numerous studies recommend using combinations of NDT techniques with DT to estimate concrete compressive strength, thereby improving the accuracy of anticipated results [13,14].

Many scholars suggested and multiple regression-analyzed a significant variety of empirical equations for calculating the compressive strength of concrete based on Ultrasonic Pulse Velocity (UPV) and Rebound Number (RN) values in the 1960s [15,16]. Regression analysis was performed on a number of experimental datasets, including power functions, bilinear functions, and biexponential functions, to derive these equations [17–19]. Furthermore, Lawson et al. [20] looked at the connection between concrete's compressive strength and UPV. With age, there is an increasing association between UPV and concrete's compressive strength; however, the amount of rise depends on the mix proportion designs and curing technique. Sbartai et al. [21] suggested a technique that utilizes an extensive database of NDT trials to obtain compressive strength and UPV data for concrete. They also carried out an assessment of the NDT methods currently used to determine the compressive strength of concrete. For a more precise evaluation of concrete strength, Ali-Benyahia et al. [2] suggested several models that bridge the gap between destructive testing and individual or combined NDTs. In comparison to current methods, the study's findings highlighted the great usefulness of the combined method in estimating concrete strength. Numerous concrete specimens were subjected to NDT for RN and UPV by Poorarbabi et al. [1]. For calculating the compressive strength of concrete, an efficient Response Surface Method (RSM) was presented, which is more precise than other existing methods. Correlation and comparison between NDT and DT methods form a pivotal aspect of materials evaluation and quality assessment [22–24]. The correlation between these testing methods lies in their shared goal of understanding material behaviours, yet their approaches are fundamentally different. NDT excels in scenarios where preserving the integrity of the tested material is crucial, offering a non-invasive means to assess properties [25]. DT, meanwhile, provides detailed and conclusive data but comes at the cost of sacrificing the specimen. This description aims to highlight the nuanced interplay between NDT and DT methods, emphasizing their complementary roles in ensuring material reliability across various industries [26]. The complex nonlinear connection between the composition of concrete and its compressive strength makes it challenging to develop precise predictive models, necessitating significant expertise and knowledge.

The development of prediction models for compressive strength provides a solution to these problems for NDT-based results. Such models not only reduce experimental effort but also minimize the expenses and time involved with standard testing procedures. The first issue with a successful prediction model is the practical dataset, as concrete is exposed to various weather and climatic conditions, which influence its strength. As a result, it is critical to create a practical dataset before making a forecast. Many studies employed various models to predict relationships obtained from DT and NDT,

which were both linear and nonlinear[27,28]. This is because concrete continues to acquire strength with age and time, and no one knows what the porosity, cement-aggregates ratio, grade of aggregate, type of cement, curing, and mixing conditions were at the time of construction. Regression analysis, a popular statistical tool, has been utilized to predict the link between different input factors and compressive strength. While linear regression is simple, the inherent nonlinearity in concrete behavior frequently makes nonlinear regression more accurate[27]. Nonetheless, these models have limitations, particularly when dealing with the complex interactions of several variables, such as porosity, cement-aggregate ratio, curing conditions, and concrete age[15].

Because different raw materials are used for concrete preparation and there are different components contribute to its strength, the linear regression model is replaced with a nonlinear regression model, which is more accurate[28]. MRA uses the least-squares fit strategy to determine the relationship between one or more independent variables and a dependent variable. However, the accuracy of regression analysis is inversely related to the number of independent variables[29]. Numerous factors contribute to the strength of concrete, and it is difficult or nearly impossible to obtain this characteristic from an existing structure[30]. In order to solve the complex relationship, one can use Machine learning algorithms such as Artificial Neural Network (ANN) and support vector machines (SVMs), which are found to be impressive and accurate in terms of results[31–34].

ANNs have developed into powerful modelling and prediction tools in a variety of engineering applications, including hydrology, meteorology, and concrete compressive strength prediction[31]. ANN's capacity to deal with nonlinearity, adapt to complicated patterns, and remain independent of raw material quality makes it an appealing option[34]. Unlike traditional regression approaches, ANN is modeled after the topology of a human brain network, allowing it to analyse related variables effectively[33]. Researchers have extensively used ANN for both quantitative and qualitative variable predictions in these various disciplines. Several researchers are actively employing artificial neural networks (ANN) to assess the compressive strength of concrete. Priyesh et al.[35] employed the cascade forward-back propagation technique to forecast concrete compressive strength; the same model and dataset were used in this research work. Prediction analysis is routinely used on concrete that has been made by substituting ingredients, but there has been little study on prediction analysis for Reinforced Cement Concrete (RCC)[34].

SVMs are gaining popularity among researchers in order to solve the complex problems of the real world[36]. SVMs are powerful supervised learning models widely used for regression and classification tasks due to their ability to handle high-dimensional data and deliver accurate results. However, the performance of an SVM model largely depends on selecting the optimal hyperparameters, including the Box Constraint, Epsilon, and Kernel Scale[37–40]. The novelty of this research work lies in that many authors have applied ANN and SVM prediction techniques for determining compressive strength, but none have focused on optimizing these techniques to increase the predictability of the model. This work builds upon the authors' previous work by Priyesh et al.[35]. In terms of using advanced optimization techniques for prediction using SVM and ANN. Only a practical dataset was used, which was generated by Priyesh et al.[35]. These parameters dictate the complexity and flexibility of the model, influencing its generalization capabilities, which helps in model generation.

The combination of SVM with optimization techniques such as Bayesian Optimization and GA represents a significant advancement in machine learning[41]. By automating the hyperparameter tuning process, these methods improve model efficiency and accuracy, reducing the need for manual intervention. Recent research has shown that models tuned using these optimization algorithms outperform standard SVM models, especially in complex real-world datasets[42]. These approaches are now widely applied in fields ranging from financial forecasting to bioinformatics, where the accurate prediction of continuous variables is critical. In machine learning, selecting these hyperparameters manually can be a daunting task, as improper tuning may lead to suboptimal performance, overfitting, or underfitting. Therefore, optimization algorithms such as Bayesian Optimization and GA are employed to automate this process and improve model accuracy[43–45]. This study explores the application of Bayesian Optimization and GA Optimization for enhancing the performance of SVM regression models.

Bayesian Optimization for SVM: Bayesian Optimization is a probabilistic model-based optimization technique

that has proven effective for hyperparameter tuning in machine learning models. It constructs a surrogate model (typically a Gaussian Process) to approximate the objective function and iteratively refines this model by selecting the most promising hyperparameters based on an acquisition function[46]. The goal is to minimize the objective function (e.g., Mean Squared Error for regression) while reducing the number of evaluations of the true objective function, making it especially suitable for computationally expensive tasks like hyperparameter tuning. Recent studies have demonstrated the efficiency of Bayesian Optimization in automating the selection of SVM hyperparameters, leading to improved model performance with fewer computational resources[43,44].

GA for SVM: GAs are inspired by the process of natural evolution and are widely used in optimization problems across various domains. GAs operate by evolving a population of candidate solutions through operators like mutation, crossover, and selection. In the context of SVM hyperparameter tuning, GA is employed to evolve potential solutions (i.e., sets of hyperparameters) over multiple generations, optimizing the model's performance iteratively. This approach is advantageous for exploring a vast search space and avoiding local minima, which traditional gradient-based methods might fall into. Several recent studies have highlighted the effectiveness of GAs in tuning SVMs, leading to better predictive accuracy and robustness in regression tasks[47]. Both optimization techniques can be used in SVM prediction. The objective of research work:

1. To find the optimal set of hyperparameters that lead to the best model performance in the prediction of the relationship between compressive strength of NDT and DT
2. To evaluate the effectiveness of these optimization strategies, the study compares the performance of a standard SVM model, an SVM optimized using Bayesian Optimization, and an SVM optimized using a genetic algorithm. The performance is assessed using metrics such as MSE, RMSE, R^2, and MAE. These metrics provide insight into the model's accuracy and its ability to generalize to unseen data.
3. Comparative analysis of various models.

2. Dataset

It is very important to create a practical dataset that represents the real-life complexities of the Concrete strength[35]. The concrete is exposed to different conditions such as high or low temperature, very or little humidity, extreme rainfall, or Drought. The strength of concrete is affected by internal and external environmental conditions. There is a need to prepare a practical dataset in order to perform a prediction model. The prediction should include the test performed on the various conditions, as well as with different grades of concrete. In the laboratory, one can get control parameters used to design a concrete mix, but in practice it is difficult to obtain such parameters accurately. That is why finally compressive strength is taken as a parameter determined by both methods, NDT and DT, which are considered. Experimental samples of various grades of concrete (M10 to M40) were prepared in the laboratory, considering 14-day and 28-day periods. Field samples were collected from a 20-years-old structure. In the laboratory, a total of 126 standard cubes (18 of each grade) prepared as a sample. Other 231 samples were not collected from the field; instead, it was collected from a 20-years-old structure, where the first NDT was performed, and then DT was carried out over it. The field data was collected from reinforced concrete columns, beams, and slabs in residential buildings. The structures were located in Bhopal, Madhya Pradesh, India, and primarily subjected to dry exposure with moderate temperature fluctuations (20–40 °C), therefore no aggressive chemical or marine environments were involved using the rebound numbers ranged from 22 to 38. In this research work, the same dataset was used, which was generated by Priyesh et al.[35].

3. Methodology

In this research work, optimization techniques used in the SVM are needed to optimize the parameters. The model generation, validation, and testing were performed using MATLAB 2024(a). The following steps are involved in model generation and its performance evaluation, as shown in **Figure 1**.

Data Preprocessing and Loading: The first step in data preprocessing is the normalization of data to organize and structure information within a database. Normalization of data is done by scaling the data in such a way that all data points fall within a range of 0 to 1. Thus, this technique

enables all data points to be brought to a common scale. The mathematical formula for normalization is given in Equation (1).

$$X = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})}, Y = \frac{(y - y_{\min})}{(y_{\max} - y_{\min})} \quad (1)$$

Where, X= Normalized value of NDT Compressive strength data set between 0 and 1; Y= Normalized value of DT Compressive strength data set between 0 and 1; x= NDT Compressive strength; y= DT Compressive strength

Then the data is loaded from an Excel file and it is further divided into training (60%), validation (30%), and test (10%) sets to train, optimize, and evaluate the model. Feature Matrix X and Target Vector Y, where, $X \in R^{n \times d}$ represent the feature matrix with n samples and d features (NDT methods used to determine the compressive strength of concrete which depends up to various factors such as age and grade), and $y \in R^n$ represents the target vector (DT obtain by compressive testing machine). Let the dataset D = $\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$, where x_i are the features and y_i is the target. Missing data handling is expressed as: D_{clean} = $\{(x_i,y_i) \in D \mid isnan(x_i) = false \wedge isnan(y_i) = false\}$

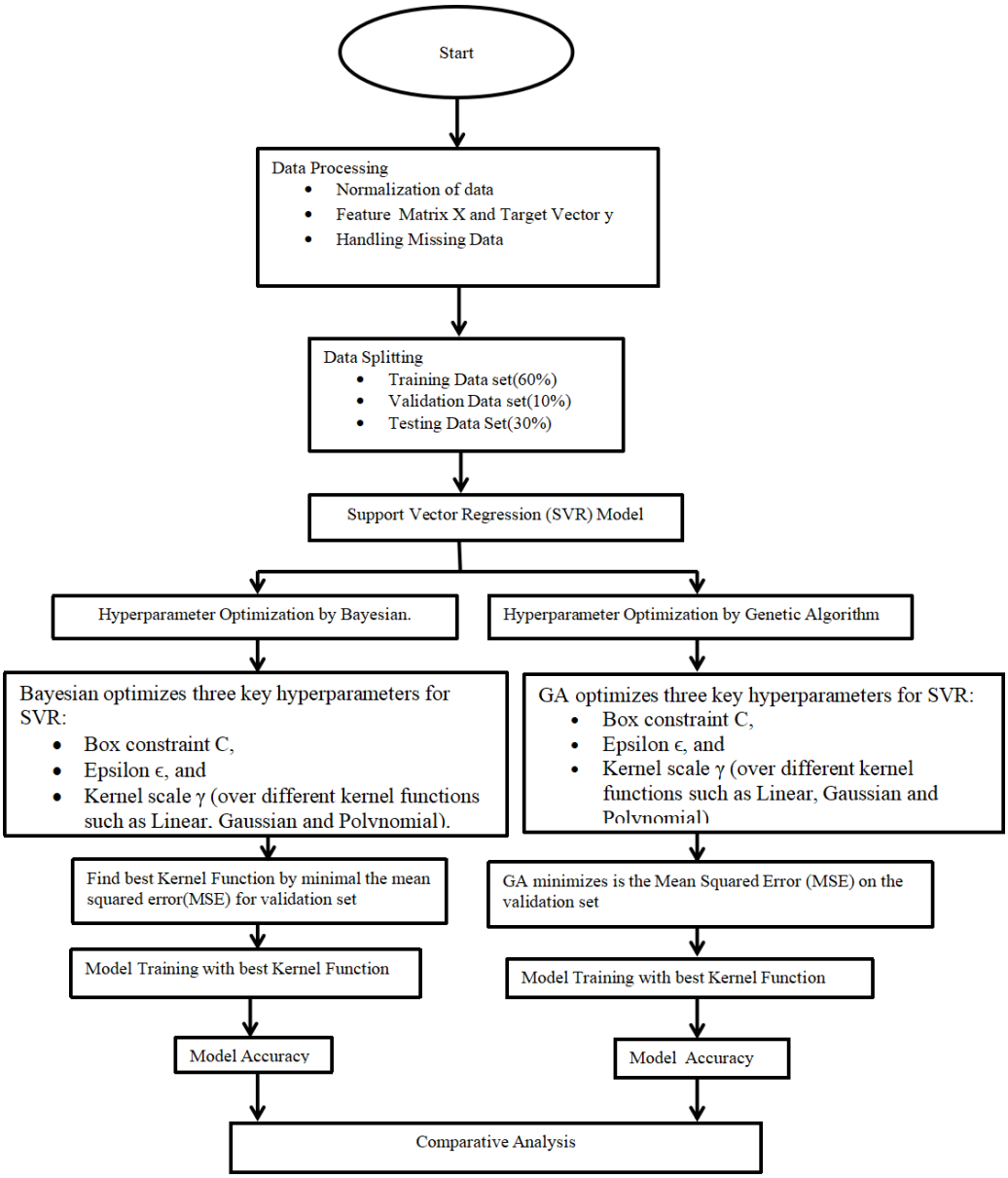


Figure 1. Methodology adopted in model generation.

Support Vector Regression (SVR) Model: The SVM objective is to build a linear regression function $f(x) = w^T X + b$ that minimizes the error in predicting y_i, Subject to constraints defined by the loss function (e.g., epsilon-insensitive loss). The problem is solved by minimizing the following objective, given in Equation (2).

$$\left\{ Min \frac{1}{2} \|W\|^2 + C \sum_{i=0}^{n} (\varepsilon_i + \xi_i^*) \right\}; y_i - \left(w^T x_i + b\right) \leq \varepsilon_i + \xi_i^*; \left(w^T x_i + b\right) - y_i \leq \varepsilon_i + \xi_i; \varepsilon_i, \xi_i^* \geq 0 \quad (2)$$

Where, W is the weight vector; b is the bias; ϵ defines the margin; ε_i, ξ_i^* are slack variables; C is the box constraint (regularization parameter)

3.1. Hyperparameter Optimization by Bayesian

When solving computationally demanding functions to locate extrema, Bayesian Optimization is a useful technique[46]. Bayesian Optimization is a probabilistic model-based optimization technique that has proven effective for hyperparameter tuning in machine learning models. It constructs a surrogate model (typically a Gaussian Process) to approximate the objective function and iteratively refines this model by selecting the most promising hyperparameters based on an acquisition function[46]. The goal is to minimize the objective function (e.g., MSE for regression) while reducing the number of evaluations of the true objective function, making it especially suitable for computationally expensive tasks like hyperparameter tuning. Recent studies have demonstrated the efficiency of Bayesian Optimization in automating the selection of SVM hyperparameters, leading to improved model performance with fewer computational resources[43,44].

Steps in Bayesian Optimization:

1. **Initialization**: Start with a small number of evaluations of the objective function (e.g., validation error). f(x) with randomly chosen hyperparameters [Equation (3)].

$$x^* = \arg \frac{\min}{x \in \chi} f(x) \quad (3)$$

Where: x= Hyperparameters (e.g., Box Constraint, Epsilon, Kernel Scale); χ = Search space of hyperparameters.

2. **Surrogate Model**: Fit a probabilistic model f(x), often a Gaussian Process, to approximate the true objective function.
3. **Acquisition Function**: Use an acquisition function a(x) to decide where to sample next. Examples include:
   - Expected Improvement (EI)
   - Probability of Improvement (PI)
   - Upper Confidence Bound (UCB)
4. **Iterative Improvement**: Evaluate the objective function at the point suggested by a(x), update the surrogate model, and repeat.

Bayesian optimization is used to optimize three SVR hyperparameters: Bayesian Optimization was employed to tune the hyperparameters, including Box Constraint, Epsilon, and Kernel Scale (Box Constraint), and (Kernel Scale) over different kernel functions[46]. The optimization aimed to minimize the error through 5-fold cross-validation.

For each kernel function k for linear [Equation (4)], Gaussian [Equation (5)] and Polynomial [Equation (6)]:

$$k_{linear}(x_i, x_j) = x_i^T x_j \quad (4)$$

$$k_{gaussian}(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2) \quad (5)$$

$$k_{polynomial}(x_i, x_j) = (x_i^T x_j + 1)^p \quad (6)$$

The goal is to minimize the mean squared error (MSE) using Bayesian Optimization for the objective function.

3.2. Hyperparameter Optimization with Genetic Algorithm (GA)

GA is a search heuristic that solves optimization issues by simulating the workings of natural selection. It evolves a population of solutions toward ideal outcomes through repetitive cycles of crossing, mutation, and selection. A genetic algorithm is generally expressed mathematically in terms of the following several crucial steps:

1. **Representation (Chromosome)**: Each of the following solutions x to the problem is usually expressed as a chromosome: a chromosome is a vector (or string) of values $x_i = (x_1, x_2, x_3 \ldots\ldots x_n)$ where x_i represents an individual in the population, and each element x_n represents a gene of the chromosome.

2. **Fitness Function**: The fitness function f(x) evaluates how good a solution is. The objective is to maximize (or minimize) this fitness function. f(X)=For maximization problems, any purpose, including optimizing performance or decreasing costs, could be represented by the fitness function.
3. **Selection**: Based on fitness, the selection process selects individuals (chromosomes) from the population for reproduction. People in good physical shape have a higher chance of being chosen. Roulette Wheel Selection is a popular method of selection in which the likelihood of choosing a particular candidate F(xi) is proportional to its fitness [Equation (7)].

$$p\left(X_i\right)=\frac{F(X_I)}{\sum_{j=1}^{N} f(x_J)} \tag{7}$$

Where N is the total population Size.

4. **Crossover(Recombination)**: Combining two parent chromosomes to create one or more offspring is known as crossover. Single-Point Crossover is a popular crossover technique.
5. **Mutation**: In order to preserve population diversity, mutations change an individual's genes, introducing randomness. For binary strings, bit-flip mutation—in which a bit (gene) is flipped with a modest probability p_{mut} is a popular mutation technique.

If a mutation takes place for a gene $x_{i,j}$, it is altered to a new value:

$$x_{i,j} \rightarrow x'_{i,j} = \text{random value (based on mutation)} \tag{8}$$

6. **New Population**: After selection, crossover, and mutation, a new population P′ is formed. The next generation is evaluated, and the process repeats.
7. **Stopping Criteria**: The algorithm terminates when a stopping condition is met, such as a maximum number of generations or when the fitness does not improve significantly. The genetic algorithm (GA) optimizes three key.

The objective function that GA minimizes is the Mean Squared Error (MSE) on the validation set: The GA optimization selects the parameters C, ϵ, γ that minimize the validation MSE for each kernel.

8. **Best Kernel Selection**: The kernel with the best performance was selected based on the minimum objective value obtained during optimization:

Model Training: The model was trained using the optimal kernel function and corresponding hyperparameters, and then predictions were made on the test set.

Model Accuracy: After model generation, 30% of the data was used for validation of this model. The validation is performed using statistical parameters such as MSE [Equation (8)], RMSE [Equation (9)], R^2 [Equation (10)], and MAE [Equation (11)].

$$MSE=\frac{\sum_1^n \left(y_i-\dot{y}_i\right)^2}{n} \tag{9}$$

$$RMSE=\sqrt{MSE} \tag{10}$$

$$R^2=1-\left(\frac{\sum_{i=1}^{n_{nest}} \left(y_i-\dot{y}_i\right)^2,}{\sum_{i=1}^{n_{nest}} \left(y_i-\overline{y_i}\right)^2}\right) \tag{11}$$

$$MAPE=\frac{1}{n}\sum_{t=1}^{n}\left|y_i-\dot{y}_i\right| \tag{12}$$

Where, y_i = Observed value; $\dot{y}_i$= Predicted value; n = Number of observations.

4. Results and Discussion

In this research, the compressive strength tests were conducted to assess the test results. The study of this experiment aims to determine the compressive strength of concrete using different test methods, such as DT and NDT, on various grades of concrete (M10 to M40) at different ages (14 and 28 days). For these total 357 samples for various the standard concrete cubes were prepared with various mix proportions that yielded standard cubes crushing strengths within a range of 10 to 40 M Pa. Combine regression of dataset 14 days, 28 days, and 2 years there were total 357 samples (70 % model generation + 30% Testing) generated with random data of 238 samples used for model creation.

4.1. Regression Model

The overall Equation (12)[35] using all grade of concrete.

$$DT=0.442+0.982NDT \tag{1}$$

Where,

DT=Compressive strength obtain Destructive Test

NDT=Compressive strength obtain Non-destructive test

Statistical data such as Multiple R, R Square, Adjusted R Square, and Standard Error, were 0.985, 0.970, 0.97, and 1.61, respectively. Therefore, there is a need to find an accurate tool and technique to find the best solution for non-linear problem.

4.2. ANN Modeling

For the ANN model formation, 238 samples are further divided into Training, Testing, and Validation. Our data set is randomly assigned from the training, testing, and validation sets containing 70% of training, 15% of Validation, and 15% for testing. The ANN result of the parameter that were used for analysis is given in **Table 1**. Coding of ANN was performed in the MATLAB 2024(Ra) such as net=newcf(input2,target2,hiddenLayerSize);

The best validation performance, as calculated by the mean square error (MSE), is obtained at 78 epochs, with an error of 0.002 as shown in **Figure 2**. The correlation coefficients for training, validation, testing, and overall are 0.987, 0.982, 0.985, and 0.986, respectively, as shown in **Figure 3**.

Table 1. Parameters used in neural network.

Parameters	Types
Number of hidden layers	1
Number of hidden neuron	5
Number of epochs	100
Transfer function of layer in hidden layer	Tan-sigmoid
Transfer function of layer in output layer	pure linear
Weight/bias learning function.	Levenberg-Marquarbt(LM) algorithm
Best linking weights and biases	Random

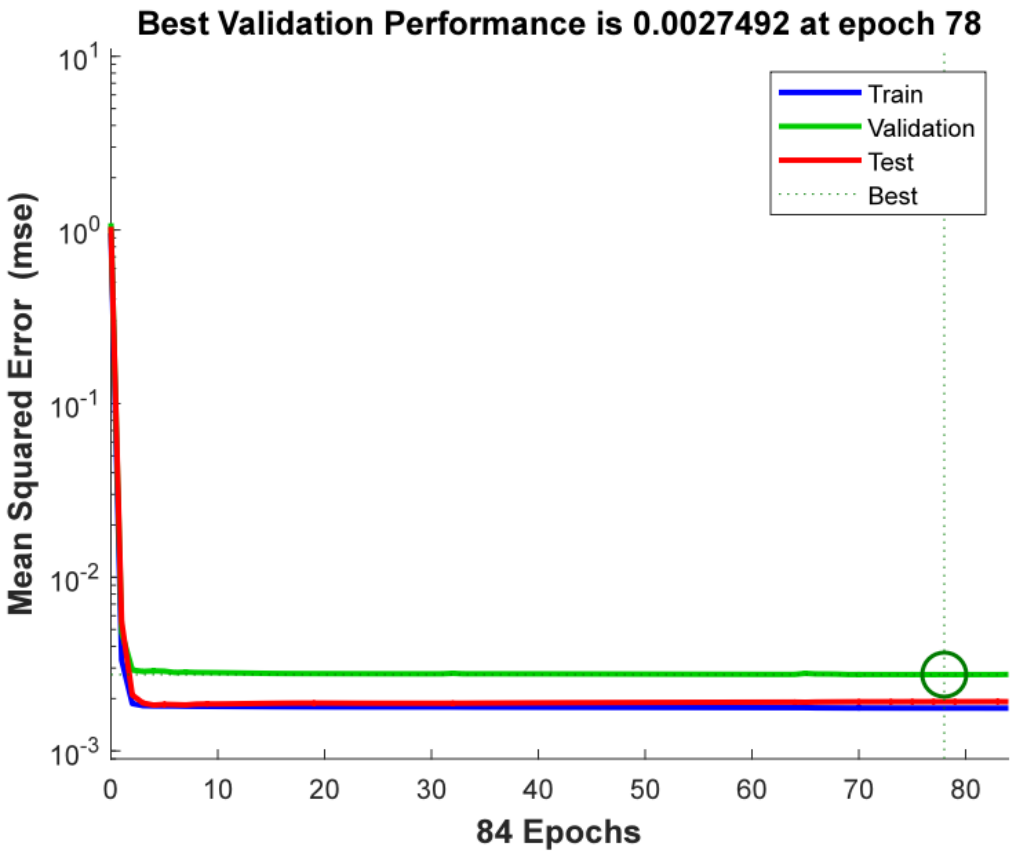


Figure 2. Best validation performance in terms mean square error[35].

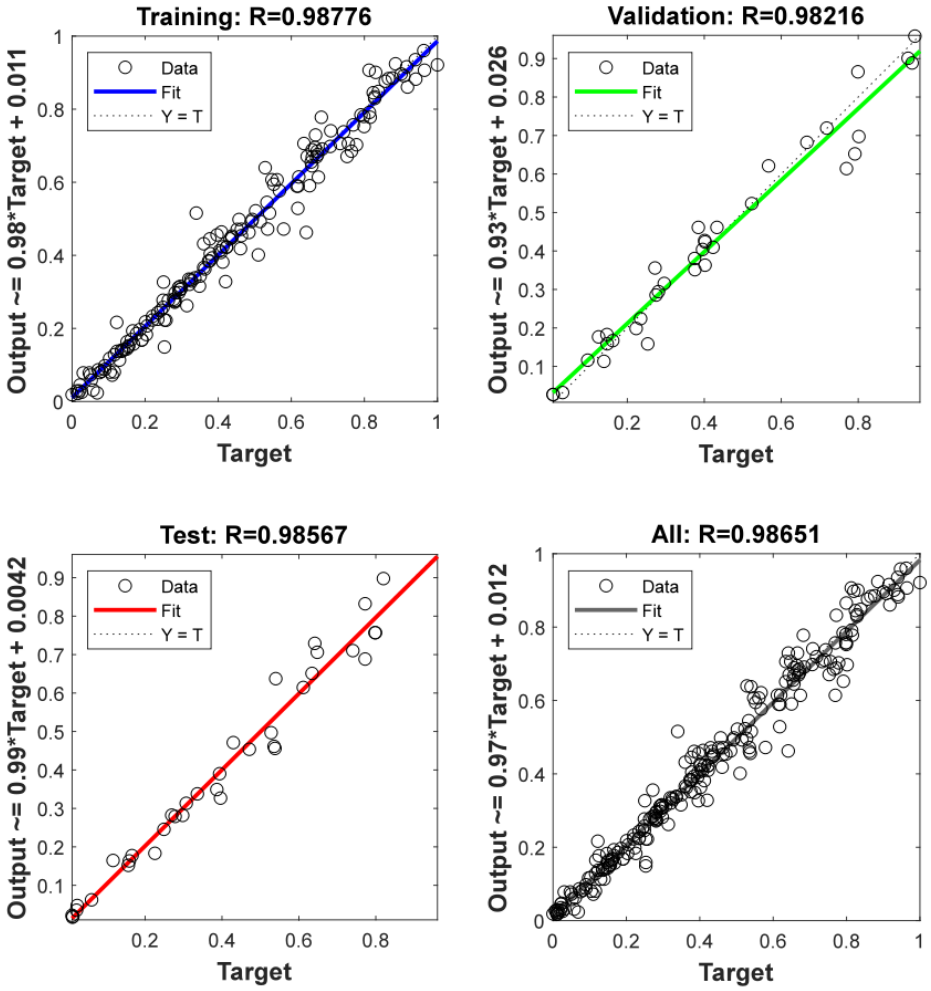


Figure 3. Correlation coefficient of training, validation, testing and overall of ANN Model [35].

4.3. SVM Models

Based on the above-mentioned parameters, prediction models such as SVM without optimization, SVM with Bayesian Optimization, and SVM with GA optimization. The results obtained are given below.

4.3.1. Result Obtain by SVM Optimization by Bayesian

Program code is written in Matlab 24R(a) code were written in library script(.m file) using function results = bayesopt(objFun, optVars,'AcquisitionFunctionName', 'expected-improvement-plus','MaxObjectiveEvaluations', 30, ...'Verbose', 0,'PlotFcn', {@plotMinObjective});

While running the Bayesian Optimization for tuning the hyperparameters of an SVR model with three different kernel functions—linear, Gaussian, and polynomial—the results obtained in **Table 2** show the difference between the minimum objective and the estimated objective for each kernel function of the models, respectively. In **Table 2**, the performance of three different kernel functions—linear, Gaussian, and polynomial—is shown, along with the corresponding optimized hyperparameters: Box Constraint, Epsilon, Kernel Scale, and MSE.

Table 2. Best estimated feasible point according to models by Bayesian Optimization.

S.No	Kernel Function	Box Constraint	Epsilon	Kernel Scale	Estimated Objective Function Value (MSE)	Estimated Function Evaluation Time
1.	linear	14.11	0.000394	0.0559	0.00214	25.4094
2.	Gaussian	15.65	0.000710	0.3014	0.00132	0.26996
3.	Polynomial	31.48	0.000340	1.8874	0.00218	0.22605

For the Linear kernel, a moderate Box Constraint value of 14.111 suggests a balance between accurately fitting the training data and ensuring generalization. The minimal Epsilon value of 0.00039457 indicates a focus on precise predictions by penalizing even minor deviations from target values. Although Kernel Scale (0.055939) has limited significance in a linear kernel, its small value may suggest that feature standardization is necessary for better performance. The Linear kernel achieves a relatively low Mean Squared Error (MSE) of 0.0021461, indicating a good fit to the data; however, the evaluation time of 25.4094 seconds is significantly higher, suggesting it is computationally intensive.

The Gaussian (RBF) kernel outperforms the others, achieving the lowest MSE of 0.0013271, which reflects its superior performance in capturing data patterns. Its Box Constraint value of 15.652 highlights an emphasis on data fitting while maintaining generalization, and the small Epsilon value of 0.00071073 ensures precise predictions. The Kernel Scale of 0.30143 is relatively small, enabling the model to focus on local patterns effectively. Additionally, the evaluation time of 0.26996 seconds is significantly lower, making the Gaussian kernel both efficient and accurate.

The Polynomial kernel, with the highest Box Constraint value of 31.485, indicates a strong focus on minimizing errors but with an increased risk of overfitting. Its minimal Epsilon value of 0.0003407 emphasizes minimizing minor prediction errors, while the larger Kernel Scale of 1.8874 allows a broader influence of each data point, capturing more complex patterns. The Polynomial kernel achieves an MSE of 0.0021841, comparable to the Linear kernel, but its evaluation time of 0.22605 seconds is the shortest among the three, making it computationally efficient.

In conclusion, the Gaussian kernel provides the best overall performance due to its low MSE and efficient evaluation time, making it the ideal choice for the dataset. The Polynomial kernel is competitive, particularly in computational efficiency and pattern complexity, but it may risk overfitting. The Linear kernel also delivers good results but is less practical due to its significantly higher computational cost.

The following results which are key points, are pointed out below:

- The Gaussian kernel has the best MSE (0.0013271), making it the optimal choice based on these results. Additionally, it has a quick evaluation time.
- Linear kernel provides a slightly higher MSE than Gaussian, but took significantly longer to evaluate. The linear kernel could be preferred in cases where model interpretability is more important than absolute performance.
- Polynomial kernel has a performance close to the linear kernel with much faster evaluation, but it is not as good as the Gaussian kernel in terms of MSE.

4.3.2. Result Obtain by SVM Optimization by GA

Program code is written in Matlab 24R(a) code were written in library script(.m file) using function objFunGA = @(params)kfoldLoss(fitrsvm(X_train, y_train,'KernelFunction', 'linear','BoxConstraint', params(1),'Epsilon', params(2),'KernelScale', 'auto','Standardize', true, 'KFold', 5)); % 5-fold cross-validation

% Run Genetic Algorithm optimization options = optimoptions('ga', 'Display', 'iter', 'PopulationSize', 30, 'MaxGenerations', 20);

The results of GA optimization for tuning the hyperparameters of an SVR model with three different kernel functions—Linear, Gaussian (RBF), and Polynomial—are presented in **Table 3**. Each kernel's performance is evaluated based on the optimized hyperparameters: Box Constraint, Epsilon, Kernel Scale, and Mean Squared Error (MSE).

Table 3. Best estimated feasible point according to models by GA.

S.No	Kernel Function	Box Constraint	Epsilon	Kernel Scale	MSE
1.	linear	95.6071	0.1213	3.5947	0.0020
2.	Gaussian	26.1893	0.0475	2.4884	0.0015
3.	polynomial	98.4047	0.0379	2.2380	0.0016

For the Linear kernel, the optimized Box Constraint value of 95.6071 is very high, indicating that the model heavily penalizes misclassifications or deviations, aiming for a tight fit to the training data with minimal tolerance for errors. The Epsilon value of 0.1213 reflects a moderate margin of tolerance for errors, allowing the model some flexibility within this range. While Kernel Scale (3.5947) is typically less critical for the linear kernel, its value suggests that input features are scaled, which in turn influences the model's interpretation of distances between data points. The MSE of 0.0020 is relatively small, showcasing good performance, although the high Box Constraint might limit flexibility for complex data.

For the Gaussian (RBF) kernel, the optimized Box Constraint value of 26.1893 is much lower compared to the linear kernel, reflecting more flexibility and tolerance for outliers during training. The smaller Epsilon value of 0.0475 indicates the model is finely tuned to reduce prediction errors. A Kernel Scale of 2.4884 allows the Gaussian kernel to capture local data patterns effectively, striking a balance between generalization and precision. This kernel achieves the lowest MSE of 0.0015 among the three, highlighting its ability to minimize prediction errors and effectively model non-linear patterns.

The Polynomial kernel features the highest Box Constraint value (98.4047), signifying a strong focus on minimizing training errors, which could lead to overfitting. The minimal Epsilon value of 0.0379 further supports this, as the model prioritizes minimizing even minor prediction errors. The Kernel Scale of 2.2380 is similar to the Gaussian kernel, enabling it to capture non-linear patterns effectively. With an MSE of 0.0016, the polynomial kernel demonstrates good performance but does not surpass the Gaussian kernel in terms of accuracy. The high Box Constraint suggests that the polynomial kernel might prioritize error minimization at the expense of flexibility.

Overall, the Gaussian kernel strikes the best balance between model complexity and prediction accuracy, making it the most suitable choice for the given dataset.

4.3.3. SVM models

The MSE results show that the Gaussian kernel yielded the lowest error in both optimization algorithms, demonstrating its effectiveness at capturing the underlying patterns in the data. This is why it performed better than the linear and polynomial kernels in your optimization results [48]. The Gaussian kernel's ability to transform the input data into a higher-dimensional feature space, while being computationally efficient, makes it often the best-performing choice in scenarios where the relationship between features is complex and non-linear for NDT and DT [49]. The Gaussian kernel, also known as the Radial Basis Function (RBF) kernel, is often preferred over other kernels like linear or polynomial in Support Vector Machine (SVM) models for several reasons:

- Non-linearity Handling: The Gaussian kernel is particularly effective in dealing with non-linear relationships between the features. Unlike the linear kernel, which assumes a linear relationship, the Gaussian kernel can model complex data distributions and capture non-linear patterns by projecting the data into a higher-dimensional space [49].
- Smoothness and Flexibility: The Gaussian kernel is very smooth, meaning it can adapt well to small changes in the data. The kernel scale (σ) controls how much influence a single data point has, giving flexibility to the model to adapt to complex decision boundaries [50].
- Generalization Capability: Gaussian kernels usually provide better generalization in many real-world problems compared to polynomial or linear kernels. It balances complexity and generalization, avoiding overfitting while capturing essential patterns in the data [50].
- Lower Sensitivity to Overfitting: The polynomial kernel, when set to higher degrees, may lead to overfitting, particularly with small datasets or noisy data. The Gaussian kernel, however, tends to be more robust, controlling overfitting by appropriately selecting the kernel width (σ) through optimization techniques like Bayesian or genetic algorithms [51].
- Wider Applicability: In practice, the Gaussian kernel has

been empirically shown to work well for a variety of problems, making it a more versatile choice for a wide range of datasets[51].

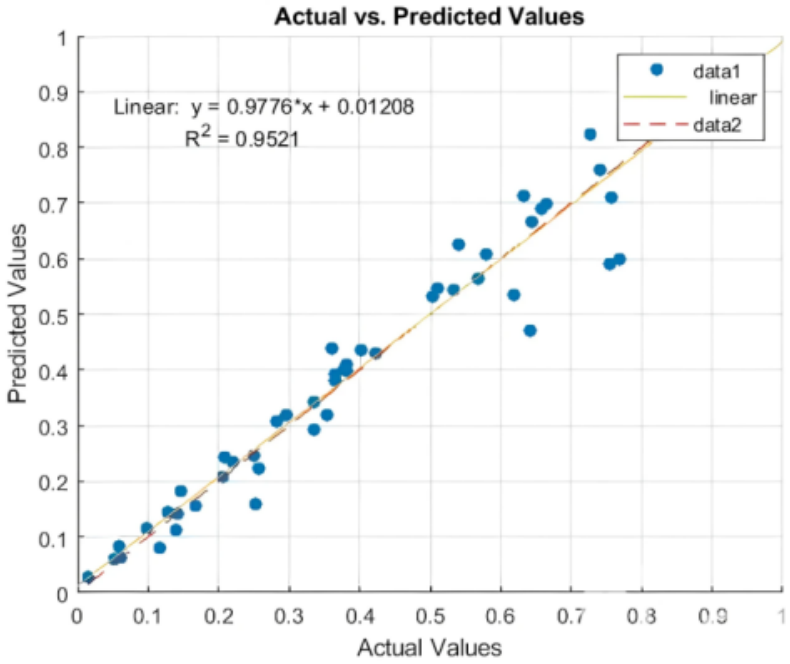
4.3.4. Validation and Comparisons of Models

The predictions of model regression analysis, ANN model, and SVM models—basic SVM (without optimization), SVM with Bayesian Optimization, and SVM with Genetic Algorithm (GA) Optimization—are made using testing data that was not used in model generation. Thirty percent (119 samples) of the total dataset were used for model validation. Statistical parameters, such as MSE, RMSE, and MAPE (%), were used as validation parameters, as shown in **Table 4**. The comparison highlights the improvement in accuracy and reduction in error achieved through the application of optimization techniques.

Table 4. Comparative analysis of various models.

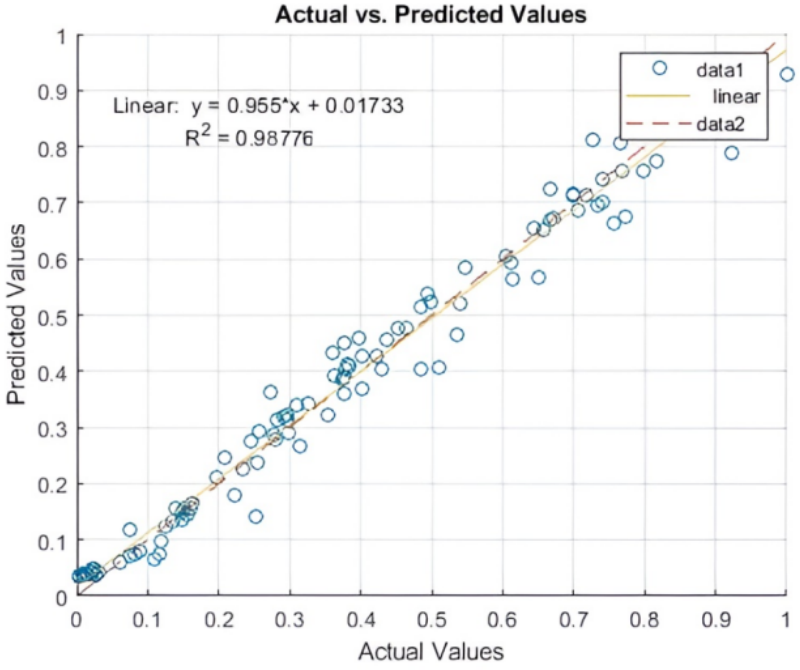
S.No	Type of Model	MSE	RMSE	MAE	R^2
1.	SVM without optimization	0.0035	0.0587	0.0452	0.952
2.	SVM with Bayesian Optimization	0.0017	0.0412	0.0310	0.987
3.	SVM with Genetic Algorithm optimization	0.0032	0.0567	0.0402	0.959
4.	ANN Model	0.0220	0.469	0.206	0.970
5.	Regression	2.9000	1.70	4.70	0.850

GA Optimization also improves the model's performance, but not as substantially as Bayesian Optimization. The MSE is reduced to 0.0032, slightly better than the basic SVM model. The RMSE decreases to 0.0567, and the MAE drops to 0.0402, showing some improvement in prediction accuracy. However, the R^2 value falls slightly to 0.9514, indicating a marginally weaker ability to explain the variance in the data. While GA optimization provides benefits, its performance is less remarkable than that of Bayesian Optimization. The comparison also includes the relationship between predicted and actual compressive strength values, as illustrated in **Figure 4** (**a**, **b**, and **c**) for linear, Gaussian, and polynomial kernels. These figures highlight that the Gaussian kernel with Bayesian Optimization achieves the best results. The Gaussian kernel's ability to model non-linear relationships between NDT and DT parameters, such as age, makes it particularly effective.

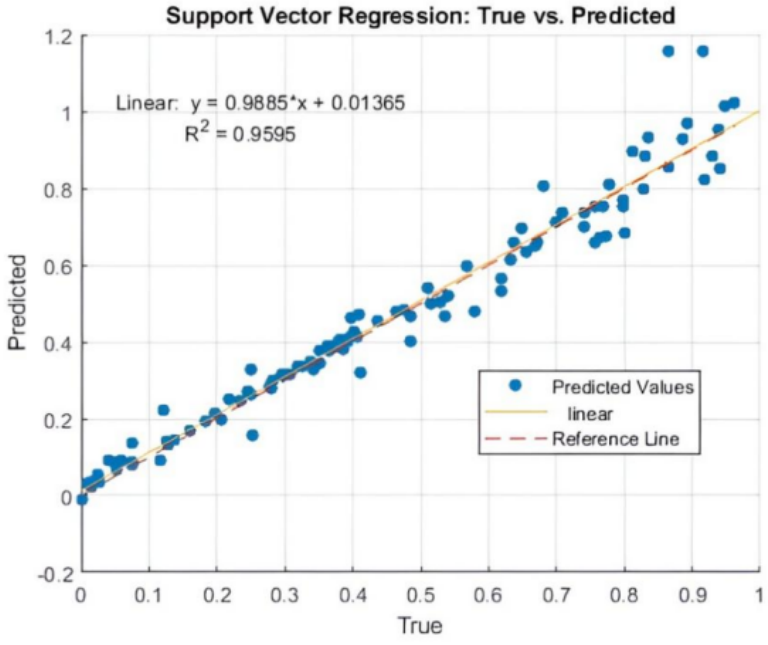


(**a**) SVM without optimization.

Figure 4. *Cont.*



(**b**) SVM with Bayesian Optimization.



(**c**) SVM with Genetic Algorithm optimization.

Figure 4. Graphical Representation of Predicated V/S Actual Value of Compressive strength. (a) SVM without optimization; (b) SVM with Bayesian Optimization; (c) SVM with Genetic Algorithm optimization.

Overall, Bayesian Optimization emerges as the most effective method for improving SVM model performance, particularly with the Gaussian kernel for non-linear data relationships. Bayesian Optimization is generally preferred over Genetic Algorithms (GA) for hyperparameters tuning in computationally expensive models like SVMs. It is faster, more sample-efficient, and better at handling noisy and expensive-to-evaluate functions due to its probabilistic modeling approach. Bayesian Optimization also excels at global optimization and targeted exploration of hyperparameters, while

GA often requires more evaluations and can get stuck in local optima.

Checking the statistical significance was done using a t-test, and the parameters of the test are given in **Table 5**. To evaluate the statistical reliability of predictions from different regression algorithms, a paired t-test was conducted comparing the predicted values against the actual test set values for each model. The results are summarized in the table, focusing on p-values, t-statistics, and 95% confidence intervals for the mean difference.

Bayesian Optimized SVR: The Bayesian Optimization-based SVR achieved a p-value of 0.8331 and a t-statistic of -0.2110, with a confidence interval of [0.0011 to 0.0231]. Since the p-value is greater than 0.05, we fail to reject the null hypothesis, indicating that there is no statistically significant difference between the predicted and actual values. This suggests that the Bayesian-optimized SVR model generalizes well and produces highly reliable predictions that align closely with ground truth values.

GA Optimized SVR: The GA (Genetic Algorithm) optimized SVR model shows a p-value of 0.03159 and a t-statistic of 2.2092, with a confidence interval that crosses zero (-0.0088 to 0.0071). Here, the p-value is less than 0.05, indicating that we reject the null hypothesis and conclude that there is a statistically significant difference between the predicted and actual values. Although GA improves performance over unoptimized models in many cases, this result implies that its optimization may not have been robust enough to yield reliable generalization on unseen data, possibly due to overfitting or local minima.

Standard (Unoptimized) SVM: The standard SVM (without any hyperparameter tuning) yielded a p-value of 0.5487, indicating no significant difference between predicted and actual outputs. While this is a positive sign of generalizability, the relatively high confidence interval spread suggests moderate prediction accuracy, but less consistency compared to Bayesian-optimized SVR. It indicates that even without tuning, SVM performs reasonably well, though not at its optimal capability.

Artificial Neural Network (ANN): The ANN model produced a p-value of 0.0185 and a t-statistic of 2.3885, with a confidence interval of [0.0017 to 0.0181]. These results clearly show a statistically significant difference between predicted and actual values, suggesting the model, despite being powerful and flexible, likely overfitted to the training data or was not tuned optimally for the regression task. This highlights the importance of regularization and careful validation when using neural networks for small or structured datasets.

Table 5. Statistical significance of various models.

S.No	Algorithm	p-Value	t-Statistic	95% Confidence Interval
1	SVM with GA optimization	0.03159	2.2092	[-0.0088 to 0.0071]
2	SVM with Bayesian Optimization	0.8331	-0.2110	[0.0011 to 0.0231]
3	SVM without optimization	0.5487	0.6016	[-0.0080 to 0.0149]
4	ANN	0.01850	2.3885	[0.0017 to 0.0181]

4.4. Discussion for Comparisons of Models

In developed countries like those in Europe, America, and Japan, the focus of construction has shifted from building new high-rise structures to maintaining and repairing existing ones. The rising costs from the deterioration of concrete structures have highlighted the importance of durability as a key quality indicator. As a result, NDT techniques are increasingly used to assess the condition and compressive strength of concrete.

Existing studies have revealed a significant need to prepare a practical and representative dataset of existing structures [35]. It involved the mixing of laboratory data with the field data set for accurate prediction. It should be prepared by a proper concrete mix design of various grades of concrete in the laboratory with different curing ages and other field data exposed under environmental conditions that the structure had undergone in due course of time. Experimental samples of various grades of concrete (M10 to M40) were prepared in the laboratory, considering 14-day and 28-day periods. Field samples were collected from a 20 years old structure. In the laboratory, a total of 126 standard cubes (18 of each grade) were prepared as a sample. Other 231 samples were collected from the field. The sample was collected from a 20-year-old structure, where the first NDT was performed, and then DT was carried out over it. The prediction of concrete compressive strength using NDT will be enhanced under a practical data set with the help of machine learning algorithms (MLAs)

such as ANN and SVM. Therefore, the integrated approach of MLA and a practical approach have given a cost-effective solution to the complex problem.

SVM is gaining more popularity among the researched due to less dataset and high accuracy is required. This study explores the SVR with different optimized models such as GA and Bayesian models. When we compare the available data and model generation, it has been found that for less number of dataset, SVM is more effective compared to the ANN model. This result might change when more datasets with complex situations are taken into account.

The SVM with Bayesian Optimization model has proved to be efficient compared to other models. The ANN model can be effective if the dataset is large, but at present, the SVM has proved to be better among them. The statistical tests are performed on SVM with Bayesian Optimization model in order to check the significance of the model. The statistical analysis conducted to evaluate the performance of the prediction model reveals that there is no significant difference between the predicted and actual values of compressive strength. A paired t-test yielded a p-value of 0.83331, which is substantially higher than the conventional significance level of 0.05. This high p-value leads to a failure to reject the null hypothesis, indicating that the observed differences between the predicted and actual values are likely due to random variation rather than any systematic error in the model.

The t-statistic value of -0.2110 is very close to zero, which suggests that the mean difference between the two sets of values is minimal. Moreover, the 95% confidence interval for the mean difference ranges from -0.0088 to 0.0071, which includes zero. This further supports the conclusion that there is no statistically significant bias in the model's predictions.

These results demonstrate that the prediction model is performing well and is capable of generating values that are statistically consistent with the actual measurements. The lack of a significant difference indicates that the model has good predictive reliability for the considered data sample. However, it is important to note that statistical non-significance does not imply that the model is perfect; instead, it suggests that within the current sample, the errors are not systematically large or directional.

To gain a more nuanced understanding of model performance, graphical tools such as residual plots (**Figure 5**) or error histograms (**Figure 6**) can be utilized. These visual diagnostics may reveal patterns such as heteroscedasticity or skewness in the error terms that are not captured by statistical tests alone. Additionally, further validation with independent datasets would enhance confidence in the model's generalizability.

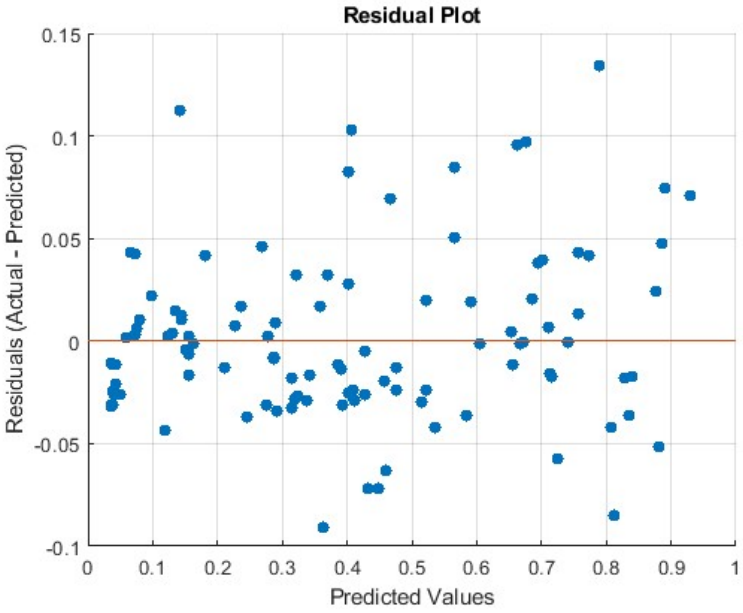


Figure 5. Residual plot between residual error and predicted value or SVM Bayesian Optimization Model.

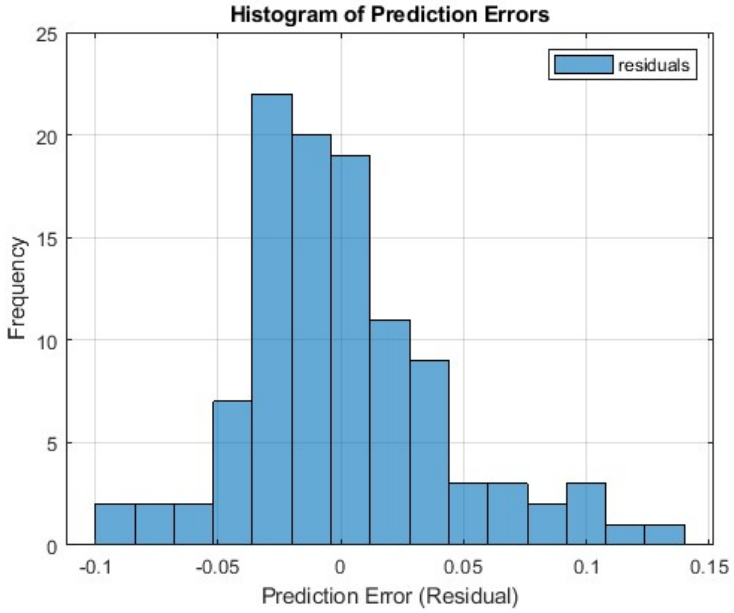


Figure 6. Histogram of predicted error of SVM Bayesian Optimization Model.

The t-test effectively highlights not only numerical performance but also the statistical reliability of model predictions, adding depth to the model selection process. Bayesian SVR outperformed all other approaches in terms of statistical closeness to actual values, then ANN and GA-based SVR exhibited statistically significant differences, indicating prediction deviations and potential overfitting. Unoptimized SVM maintained moderate performance and acceptable generalization, making it a stable baseline.

While promising, the study is limited by the relatively small dataset of 357 samples. Expanding the dataset and incorporating advanced techniques in future research will further enhance the predictive accuracy and practical applicability of these models for evaluating concrete structures. The following are limitations and suggestions for future work:

Data Availability: The limited dataset used for model training and validation may have introduced biases, which could potentially impact the reliability of the results. Future studies should aim to collect more diverse datasets, encompassing a wide range of concrete grades, ages, and environmental conditions.

Field Variability: Variations in site conditions, such as surface orientation (vertical or horizontal), moisture content, and testing instrument accuracy, may affect NDT measurements and, consequently, the prediction accuracy. These factors should be standardized or explicitly accounted for in future models.

Model Complexity: ANN models, while accurate, can be computationally intensive and may require parameter tuning for optimal performance. This increases the complexity of their application in real-world scenarios compared to regression models. The incorporation of future Optimization algorithms, such as the Whale Optimization Algorithm and the Gray Wolf Optimization Algorithm, could be employed to fine-tune model parameters.

Integration of Hybrid Models: Hybrid approaches combining regression, ANN, and heuristic optimization techniques can provide a balanced solution, leveraging the strengths of each method.

Sensitivity Analysis: Future studies should perform sensitivity analyses to identify the most influential factors affecting compressive strength prediction and refine the models accordingly.

Real-Time Applications: Incorporating real-time NDT data collection and model predictions into construction monitoring systems could provide on-site assessments of structural integrity.

While this study underscores the potential of ML in accurately predicting concrete compressive strength, its success hinges on the availability of extensive and diverse datasets. Expanding the dataset, addressing field variability, and leveraging advanced computational methods will pave the way for more reliable and practical predictive models. Future work should focus on these aspects to establish a standard-

ized methodology for compressive strength prediction in structural engineering.

5. Conclusion

These results are based on the available data and the best of our knowledge by exploring the relationship between NDT and DT. There were 357 samples of concrete specimens with various mix proportion designs and curing age,s along with onsite data collection and preparation in the Lab. These were used to explore the complex behaviour. The following Conclusion can be drawn:

1. SVM is more suitable when the dataset is small. These results can be changed when more complexities and higher datasets are compared to ANN and other models.
2. The prediction accuracy will depend upon the type of model used and the optimization techniques used to determine it.
3. The Gaussian non-linear kernel function is suitable for optimization techniques such as Bayesian and Genetic Algorithms, compared Linear and polynomial kernel functions. In both algorithms, the MSE is minimum, making it both efficient and effective (0.0013271). Other parameters, such as Box Constraint values, Epsilon, and Kernel Scale values, also support the result.
4. Bayesian Optimization with Gaussian non-linear kernel function provides the most significant improvement in model performance compared to other models.

Author Contributions

Conceptualization, A.K.P.; methodology, A.K.P.; formal analysis, P.G.; investigation, P.G.; data curation, P.G.; writing—original draft preparation, P.G.; writing—review and editing, A.K.P.; supervision, A.K.P. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

The authors would like to thank Krishna Consultant, the Supervision Quality Control Agency, for providing technical and equipment support. Special thanks are extended to Prof. Vijay K. Agrawal, Vice Chancellor of RKDF University, Madhya Pradesh, Bhopal, for providing necessary research facilities and for inspiring the initial steps in research development, alongside the advisor's continued guidance.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Poorarbabi, A., Ghasemi, M., Moghaddam, M.A., 2020. Concrete compressive strength prediction using non-destructive tests through response surface methodology. Ain Shams Engineering Journal. 11(4), 939–949. DOI: https://doi.org/10.1016/j.asej.2020.02.009

[2] Ali-Benyahia, K., Sbartai, Z., Breysse, D., et al., 2019. Improvement of nondestructive assessment of on-site concrete strength: influence of the selection process of cores location on the assessment quality for single and combined NDT techniques. Construction and Building Materials. 195, 613–622. DOI: https://doi.org/10.1016/j.conbuildmat.2018.10.032

[3] Dangi, V.P., Sohani, S.K., Patel, A.K., 2025. Experimental investigation on the utilization of e-waste, plastic waste, and crushed coconut shell as partial replacements for natural aggregates in concrete. Journal of Integrated Science and Technology. 13(5), 1102. DOI: https://doi.org/10.62110/sciencein.jist.2025.v13.1102

[4] Asteris, P.G., Skentou, A.D., Bardhan, A., et al., 2021. Soft computing techniques for the prediction of concrete compressive strength using nondestructive tests. Construction and Building Materials. 303, 124450. DOI: https://doi.org/10.1016/j.conbuildmat.2021.124450

[5] Bilgehan, M., 2011. A comparative study for the con-

crete compressive strength estimation using neural network and neuro-fuzzy modelling approaches. Nondestructive Testing and Evaluation. 26(1), 35–55.

[6] Kaboosi, K., Kaboosi, F., Fadavi, M., 2020. Investigation of greywater and zeolite usage in different cement contents on concrete compressive strength and their interactions. Ain Shams Engineering Journal. 11(1), 201–211. DOI: https://doi.org/10.1016/j.asej.2019.08.008

[7] Poorarbabi, A., Ghasemi, M., Moghaddam, M.A., 2020. Concrete compressive strength prediction using neural networks based on non-destructive tests and a self-calibrated response surface methodology. Journal of Nondestructive Evaluation. 39, 784. DOI: https://doi.org/10.1007/s10921-020-00718-w

[8] Karahan, S., Buyuksarac, A., Isik, E., 2020. The relationship between concrete strengths obtained by destructive and non-destructive methods. Iranian Journal of Science and Technology, Transactions of Civil Engineering. 44(1), 91–105. DOI: https://doi.org/10.1007/s40996-019-00334-3

[9] Bogas, J.A., Gomes, M.G., Gomes, A., 2013. Compressive strength evaluation of structural lightweight concrete by non-destructive ultrasonic pulse velocity method. Ultrasonics. 53(5), 962–972. DOI: https://doi.org/10.1016/j.ultras.2012.12.012

[10] Mata, R., Ruiz, R.O., Nunez, E., 2023. Correlation between compressive strength of concrete and ultrasonic pulse velocity: a case of study and a new correlation method. Construction and Building Materials. 369, 130569. DOI: https://doi.org/10.1016/j.conbuildmat.2023.130569

[11] Rashid, K., Waqas, R., 2017. Compressive strength evaluation by non-destructive techniques: an automated approach in construction industry. Journal of Building Engineering. 12, 147–154. DOI: https://doi.org/10.1016/j.jobe.2017.05.010

[12] Pucinotti, R., 2015. Reinforced concrete structure: nondestructive in situ strength assessment of concrete. Construction and Building Materials. 75, 331–341. DOI: https://doi.org/10.1016/j.conbuildmat.2014.11.023

[13] Breysse, D., 2012. Nondestructive evaluation of concrete strength: an historical review and a new perspective by combining NDT methods. Construction and Building Materials. 33, 139–163. DOI: https://doi.org/10.1016/j.conbuildmat.2011.12.103

[14] Yilmaz, N.G., Goktan, R.M., 2019. Comparison and combination of two NDT methods with implications for compressive strength evaluation of selected masonry and building stones. Bulletin of Engineering Geology and the Environment. 78(6), 4493–4503. DOI: https://doi.org/10.1007/s10064-018-1382-7

[15] Erdal, M., 2009. Prediction of the compressive strength of vacuum processed concretes using artificial neural network and regression techniques. Scientific Research and Essays. 4(10), 1057–1065.

[16] Huang, Q., Gardoni, P., Hurlebaus, S., 2011. Predicting concrete compressive strength using ultrasonic pulse velocity and rebound number. Acids Materials Journal. 108(4), 403–412.

[17] Amini, K., Jalalpour, M., Delatte, N., 2016. Advancing concrete strength prediction using non-destructive testing: development and verification of a generalizable model. Construction and Building Materials. 102(1), 762–768. DOI: https://doi.org/10.1016/j.conbuildmat.2015.10.131

[18] Shariati, M., Ramli-Sulong, N.H., Arabnejad, M.M.K.H., et al., 2011. Assessing the strength of reinforced concrete structures through ultrasonic pulse velocity and Schmidt rebound hammer tests. Scientific Research and Essays. 6(1), 213–220.

[19] Trtnik, G., Kavcic, F., Turk, G., 2009. Prediction of concrete strength using ultrasonic pulse velocity and artificial neural networks. Ultrasonics. 49(1), 53–60. DOI: https://doi.org/10.1016/j.ultras.2008.05.001

[20] Lawson, I., Danso, K., Odoi, H., 2011. Non-destructive evaluation of concrete using ultrasonic pulse velocity. Research Journal of Applied Sciences, Engineering and Technology. 6(3), 499–504.

[21] Sbartai, Z., Breysse, D., Larget, M., et al., 2012. Combining NDT techniques for improved evaluation of concrete properties. Cement and Concrete Composites. 34(6), 725–733. DOI: https://doi.org/10.1016/j.cemconcomp.2012.03.005

[22] Aydın, F., Sarıbıyık, M., 2010. Compressive strength of various ages concretes by non-destructive test methods. Scientific Research and Essays. 5, 1644–1648.

[23] Malek, J., Kaouther, M., 2014. Destructive and non-destructive testing of concrete structures. Jordan Journal of Civil Engineering. 8(4), 432–441.

[24] Samson, D., Moses, O.T., 2014. Correlation between non-destructive testing (NDT) and destructive testing (DT) of compressive strength of concrete. International Journal of Engineering Science Invention. 3(9), 12–17.

[25] Aydin, F., Saribiyik, M., 2010. Correlation between Schmidt Hammer and destructive compressions testing for concretes in existing buildings. Scientific Research and Essays. 5(13), 1644–1648.

[26] Abdullah, B., Abdulkadir, M., 2016. Correlation between destructive and non-destructive tests results for concrete compressive strength. Journal of Zankoy Sulaimani. 18(4), 119–132.

[27] Güçlüer, K., Özbeyaz, A., Göymen, S., et al., 2021. A comparative investigation using machine learning methods for concrete compressive strength estimation. Materials Today Communications. 27, 102278. DOI: https://doi.org/10.1016/j.mtcomm.2021.102278

[28] Kewalramani, M.A., Gupta, R., 2006. Concrete compressive strength prediction using ultrasonic pulse velocity through artificial neural networks. Automation

in Construction. 15(3), 374–379.

[29] Abhilash, P.T., Satyanarayana, P.V.V., Tharani, K., 2021. Prediction of compressive strength of roller compacted concrete using regression analysis and artificial neural networks. Innovative Infrastructure Solutions. 6, 218. DOI: https://doi.org/10.1007/s41062-021-00590-1

[30] Tsivilis, S., Batis, G., Chaniotakis, E., et al., 2000. Properties and behavior of limestone cement concrete and mortar. Cement and Concrete Research. 30(10), 1679–1683.

[31] Patel, A., Katiyar, S.K., Prasad, V., 2016. Bathymetric mapping of Bhopal City Lower Lake using IRS-P6: LISS-4 imagery and artificial neural network technique. Journal of the Indian Society of Remote Sensing. 44, 605–616. DOI: https://doi.org/10.1007/s12524-015-0523-8

[32] Asteris, P.G., Skentou, A.D., Bardhan, A., et al., 2021. Soft computing techniques for the prediction of concrete compressive strength using non-destructive tests. Construction and Building Materials. 303, 124450. DOI: https://doi.org/10.1016/j.conbuildmat.2021.124450

[33] Chithra, S., Kumar, S.S., Chinnaraju, K., et al., 2016. A comparative study on the compressive strength prediction models for high performance concrete containing nano silica and copper slag using regression analysis and artificial neural networks. Construction and Building Materials. 114, 528–535. DOI: https://doi.org/10.1016/j.conbuildmat.2016.03.214

[34] Zhang, W., Liu, D., Cao, K., 2024. Prediction of concrete compressive strength using support vector machine regression and non-destructive testing. Case Studies in Construction Materials. 21, e03416. DOI: https://doi.org/10.1016/j.cscm.2024.e03416

[35] Priyesh, G., Patel, A.K., 2025. Prediction of compressive strength by considering practical consideration non-destructive test by artificial neural network. Journal of Materials and Engineering Structures. 12(1), 47–62.

[36] Zhang, W., Liu, D., Cao, K., 2024. Prediction of concrete compressive strength using support vector machine regression and non-destructive testing. Case Studies in Construction Materials. 21, e03416. DOI: https://doi.org/10.1016/j.cscm.2024.e03416

[37] Roy, A., Chakraborty, S., 2023. Support vector machine in structural reliability analysis: a review. Reliability Engineering and System Safety. 233, 109126. DOI: https://doi.org/10.1016/j.ress.2023.109126

[38] Dishaa, 2024. Introduction support vector machines (SVM) with Python implementation. Available from: https://www.analyticsvidhya.com/blog/2021/04/insight-into-svm-support-vector-machine-along-with-code/ (cited 18 October 2024).

[39] Viswa, 2024. Support vector regression: unleashing the power of non-linear predictive modeling. Available from: https://medium.com/@vk.viswa/support-vector-regression-unleashing-the-power-of-non-linear-predictive-modeling-d4495836884 (cited 18 October 2024).

[40] Chen, X.X., Huang, J., 2019. Odor source localization algorithms on mobile robots: A review and future outlook. Robotics and Autonomous Systems. 112, 123–136.

[41] Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical Bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems. 25, 2951–2959.

[42] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Information Sciences. 179(13), 2232–2248.

[43] Elshewey, A.M., Shams, M.Y., El-Rashidy, N., et al., 2023. Bayesian optimization with support vector machine model for Parkinson disease classification. Sensors. 23(4), 2085. DOI: https://doi.org/10.3390/s23042085

[44] Frazier, P.I., 2018. A tutorial on Bayesian optimization. arXiv preprint. arXiv:1807.02811.

[45] Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley: Boston, MA, USA.

[46] Shahriari, B., Swersky, K., Wang, Z., et al., 2016. Taking the human out of the loop: a review of Bayesian optimization. Proceedings of the IEEE. 104(1), 148–175.

[47] Mirjalili, S., 2019. Genetic Algorithm. In: Evolutionary Algorithms and Neural Networks. Springer: Cham, Switzerland. pp. 43–55.

[48] Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer: New York, NY, USA.

[49] Schölkopf, B., Smola, A.J., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press: Cambridge, MA, USA.

[50] Keerthi, S.S., Lin, C.J., 2003. Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation. 15(7), 1667–1689.

[51] Chang, C.C., Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2(3), 1–27.