AI-based KNN Approaches for Predicting Cooling Loads in Residential Buildings

Zhaofang Du
Henan Industry and Trade Vocational College, Zhengzhou Henan, 450053, China

***Abstract*—Cooling Load (CL) estimation in residential buildings is crucial for optimizing energy consumption and ensuring indoor comfort. This article presents an innovative approach that leverages Artificial Intelligence (AI) techniques, particularly K-Nearest Neighbors (KNN), in combination with advanced optimizers, including Dynamic Arithmetic Optimization (DAO) and Wild Geese Algorithm (WGA), to enhance the accuracy of CL predictions. The proposed method harnesses the power of KNN, a machine-learning algorithm renowned for its simplicity and efficiency in regression tasks. By training on historical CL data and relevant building parameters, the KNN model can make precise predictions, 768 sample with considering factors such as Glazing Area, Glazing Area Distribution, Surface Area, Orientation, Overall Height, Wall Area, Roof Area, and Relative Compactness. Two state-of-the-art optimizers, DAO and WGA, are introduced to refine the CL estimation process further. The integration of KNN with DAO and WGA yields a robust AI-driven framework proficient in the precise estimation of CL in residential constructions. This approach not only enhances energy efficiency by optimizing cooling system operations but also contributes to sustainable building design and reduced environmental impact. Through extensive experimentation and validation, this study demonstrates the effectiveness of the proposed method, showcasing its potential to revolutionize CL estimation in residential buildings. The results indicate that the hybridization of KNN with DAO optimizers yields promising outcomes in predicting CL. The high R2 value of 0.996 and low RMSE value of 0.698 demonstrate the accuracy of the KNDA model.**

Keywords*—Cooling load; K-nearest neighbor; dynamic arithmetic optimization; wild geese algorithm

I. Introduction

In an era marked by burgeoning concerns over energy efficiency and environmental sustainability, the demand for more innovative and precise methods of managing cooling loads (CL) in residential buildings has never been more pressing [1]. Achieving the delicate balance between maintaining indoor comfort and minimizing energy consumption is a multifaceted challenge that resonates with homeowners and the broader global community [2]. The need to develop innovative approaches to predict, control, and optimize cooling loads is paramount, and this article delves into the forefront of these advancements [3]. Residential buildings constitute a substantial portion of global energy consumption [4]. Cooling systems, essential for creating comfortable living environments, contribute significantly to this energy usage [5]. CL management inefficiency can lead to excessive energy consumption, elevated utility bills, and increased carbon emissions. Hence, the stakes are high, both economically and environmentally, in devising strategies that can predict and optimize cooling loads with unparalleled accuracy [6].

Precisely forecasting building energy consumption represents a crucial aspect of energy modeling. Yet, it frequently struggles to provide a comprehensive reflection of real-world performance [7], [8]. Conventional energy models, well-suited for initial assessments, rely on engineering calculations grounded in physical principles to gauge building energy consumption [9]. Multiple research investigations have highlighted the significant gap between these forecasts and actual energy usage, sometimes surpassing the predictions by a factor of two or three. Numerical simulation techniques address these constraints when simulating building energy usage [10]. However, their capacity to accurately replicate the intricacies of the actual world remains limited. Through a systematic review of past research findings and limitations, these simulations play a pivotal role in tackling the challenges linked to using machine learning models to enhance building energy efficiency [11].

Artificial Intelligence, particularly Machine Learning (ML) [12], has emerged as a potent tool for addressing complex challenges across various domains. In the context of cooling load estimation, ML algorithms shine as they have the capacity to assimilate vast datasets encompassing diverse parameters such as outdoor temperatures [13], humidity levels, occupancy patterns, and architectural features. Among the myriad of ML algorithms, the K-Nearest Neighbors (KNN) algorithm stands out for its simplicity and effectiveness in regression tasks [14]. KNN operates on the premise that similar data points in a feature space tend to have similar output values [15]. Leveraging this principle, KNN can predict cooling loads by identifying neighboring data points with known CL values. The algorithm computes weighted averages of these neighbors, providing an accurate CL estimate based on the historical data [16]. The application of KNN in cooling load estimation is a cornerstone of this article, offering a foundation upon which advanced optimization techniques can be built [17].

In order to accurately capture intricate energy consumption patterns, the system harnessed the capabilities of a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network. Kim and Cho [18] addressed the challenge of accurately predicting housing energy consumption in the context of a rapidly increasing human population and technological development. The authors proposed a CNN-LSTM neural network, combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), to effectively predict energy consumption. The method

demonstrated nearly perfect prediction performance, outperforming conventional forecasting methods and achieving the smallest root mean square error, especially for datasets on individual household power consumption. Empirical analyses of variables provided valuable insights into the factors influencing power consumption forecasts, contributing to improved prediction accuracy. Moradzadeh et al. [19] focused on applying SVR and MLP models to estimate cooling and heating demands. The MLP model produced remarkable results for their investigation with an amazing R2-value of 0.9993 for heating load prediction, while the SVR model performed very well with an R-value of 0.9878 for cooling load prediction. These results show the degree of accuracy that a machine learning system is capable of achieving. In study [20], using a genetic algorithm in conjunction with a dynamic simulation tool, a multi-objective optimization was carried out to improve energy efficiency in existing buildings via renovations and HVAC systems. A different research suggested using statistical analysis in energy forecasting to cool an office building. The study in [21] proposed four hybrid methods for predicting the cooling load efficiency of buildings, based on artificial neural networks (ANN) and meta-heuristic algorithms such as artificial bee colony (ABC), particle swarm optimization (PSO), imperialist competitive algorithm (ICA), and genetic algorithm (GA). Cooling load forecasting was executed in study [22] using a probabilistic entropy-based neural ($PENN$) method. Short-term cooling load prediction, aiming to optimize $HVAC$ systems and energy efficiency in buildings, was performed in [23] using multiple nonlinear regression (MNR), auto-regressive (AR), and autoregressive with exogenous (ARX) models. In study [24], a feedforward neural network ($FFNN$) reduced building energy consumption for thermal comfort by 36.5%. For energy demand forecasting and energy efficiency measures in a residential building, [25] suggested a decision tree method. A comparative study of cooling load forecasting methods was conducted in study [26], contrasting machine learning methods such as minimax probability machine regression (MPMR), gradient boosted machine (GBM), deep neural network (DNN), and Gaussian process regression (GPR). In research [27], ANN, categorization and regression tree (CART), general linear regression (GLR), and chi-squared automatic interaction detector (CHAID) were used to forecast the cooling loads of the building. The networks' inputs for the prediction were the technical parameters of the building [28].

Conversely, in the context of CL prediction, the SVR model outperformed, achieving the highest $R-$value of 0.9878. In a separate study, Roy et al. [3] presented a customized Deep Neural Network (DNN) model designed to accurately anticipate heating and cooling needs in residential buildings. The results demonstrated that when it came to heating and cooling load prediction, the DNN and GPR models achieved the maximum Variance Accounted For (VAF). In the next stage of the study, the DNN model's performance was contrasted with that of the gradient-boosted machine (GBM), Gaussian process regression (GPR), and Minimax Probability Machine Regression ($MPMR$) models.

This study makes a significant contribution to the field of building energy efficiency by delving into the innovative integration of Artificial Intelligence (AI) and advanced optimization techniques for the prediction and optimization of cooling loads in residential buildings. The core innovation lies in the utilization of the K-Nearest Neighbors (KNN) base model, chosen for its efficiency and reliability in predicting building cooling loads. To further enhance the performance of the KNN model, the study introduces a novel hybridization technique that integrates two cutting-edge optimizers: Dynamic Arithmetic Optimization (DAO) and the Wild Geese Algorithm (WGA). This hybrid approach aims to harness the strengths of both optimizers, maximizing predictive accuracy and optimizing cooling load outcomes. The study's distinctive contribution lies in its comprehensive examination of various models, including individual configurations of KNN, DAO, and WGA, as well as their hybrid combinations. This meticulous evaluation ensures an unbiased assessment of each model's capabilities, providing valuable insights into their standalone and synergistic performances. Crucially, the study emphasizes the use of established metrics such as R^2 (coefficient of determination) and RMSE (root mean square error) in evaluating model performance. By incorporating these metrics, the research ensures a robust and credible assessment of the predictive capabilities of the models. This study not only explores the potential of AI and optimization techniques in enhancing energy efficiency but also establishes a methodological framework for evaluating and implementing these technologies in the context of residential building cooling load prediction.

In the following sections, a detailed examination of the relevant data, the model, and the optimizers utilized in Section II will be undertaken. An elaborate explanation of the data and an assessment of the models based on metrics will be provided. In Section III, the results derived from the training and testing phases will be scrutinized, and subsequently, the performance of the models based on classification will be reported. Finally, in Section IV, conclusions regarding the study in question and the overall performance of the models will be presented.

II. Materials and Methodology

A. Data Gathering

This article delves into the crucial parameters and variables pertinent to studying building energy consumption, particularly in predicting cooling loads. The dataset is meticulously divided into three segments: Training (70%), Validation (15%), and Testing (15%). Each segment plays a pivotal role in different phases of model development and assessment. The Training Set forms the foundation for training the predictive model, enabling it to learn from historical data. The Validation Set fine-tunes the model's parameters, guarding against overfitting and ensuring robustness. Finally, the Testing Set rigorously evaluates the model's efficacy with unseen data, providing the ultimate assessment. These parameters are essential for comprehending and modeling energy dynamics in residential buildings. Table I presents the statistical characteristics of the input variables [29]. The following is a detailed breakdown of each parameter:

1) Relative compactness: Relative compactness is a fundamental parameter that describes how tightly or

efficiently a building is designed. It is a dimensionless value that quantifies the compactness of the building's shape, affecting the thermal performance and energy consumption.

2) Surface area: Surface area is critical as it directly influences the heat exchange between the building's interior and the external environment. It encompasses the total external surface area of the building, which includes walls, roof, and possibly other exposed surfaces.

3) Wall area: The Wall represents the total surface area of the building's walls. Walls are significant in heat transfer and insulation, making this parameter crucial for energy modeling.

4) Roof area: The roof area is the total surface area of the building's roof. Roof design and insulation are key factors affecting cooling load, as heat gain through the roof can be substantial.

5) Overall height: The height of the building impacts its internal volume and air circulation, influencing the distribution of cooling loads within the structure.

6) Orientation: Building orientation refers to the direction in which the building faces. It can affect the solar radiation the building receives, impacting the cooling load.

7) Glazing area: The glazing area represents the proportion of the building's external envelope covered by windows or glass. It significantly influences heat gain and loss, making it an essential factor in cooling load calculations.

8) Glazing area distribution: The distribution of glazing within the building's envelope can vary, affecting how heat is distributed and the spatial variations in cooling load.

9) Cooling: Cooling load in kilowatts (KW) represents the cooling energy required to maintain a comfortable indoor temperature. It is a crucial output variable in energy modeling.

These parameters collectively serve as the cornerstone for predicting cooling loads in residential buildings. Fig. 1 illustrates the correlation matrix depicting relationships among the input and output variables. The article delves into examining the influence of these parameters on energy consumption and explores how advanced machine learning models, like KNN integrated with innovative hybridization techniques, can enhance the precision of cooling load predictions. Understanding these material factors is essential for optimizing energy-efficient building design and cooling system operation [30].

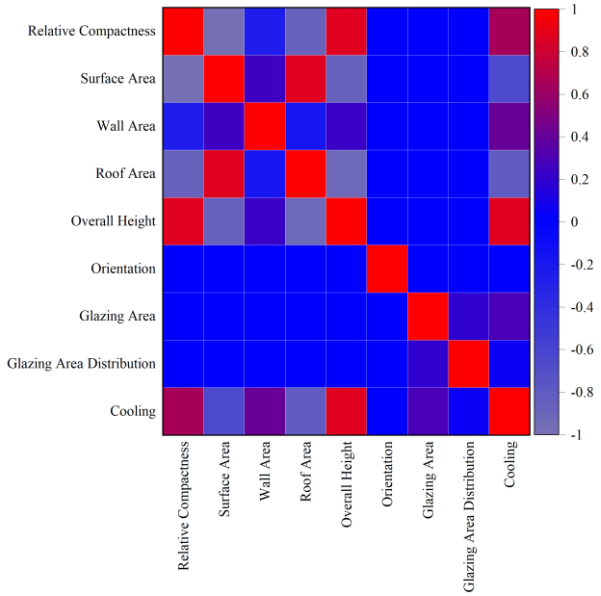


Fig. 1. Correlation matrix for the input and output variables.

TABLE I. THE STATISTIC PROPERTIES OF THE INPUT VARIABLE OF KNN

Variables	Indicators				
	Category	***Min***	***Max***	***Avg***	***St. Dev.***
Relative compactness	*Input*	0.62	0.98	0.764	0.106
Surface area (m2)	*Input*	514.5	808.5	671.7	88.09
Wall area (m2)	*Input*	245	416.5	318.5	43.63
Roof area (m2)	*Input*	110.3	220.5	176.6	45.17
Overall height (m)	*Input*	3.5	7	5.25	1.751
Orientation	*Input*	2	5	3.5	1.119
Glazing area (%)	*Input*	0	0.4	0.234	0.133
Glazing area distribution	*Input*	0	5	2.813	1.551
Cooling (KW)	*Output*	6.01	43.1	22.31	10.09

B. KNN-based

The K-Nearest Neighbors (KNN) algorithm predicts outcomes by considering most feedback from *K* data points closest to the test point [31]. To prepare for the application of this algorithm, it is crucial to standardize these parameters using Eq. (1).

$$x_{normalization} = \frac{x - Min}{Max - Min} \quad (1)$$

Next, the Euclidean distance between the test and data points is determined using Eq. (2).

$$H(x_i, x_j) = \left(\sum_{h=1}^{m} \left|x_i^{(h)} - x_j^{(h)}\right|^2\right)^{\frac{1}{2}} \quad (2)$$

Eq. (2) calculates the Euclidean distance H, where m is the number of argument points, between the test point (x_j) and the original data points (x_i). But since different parameters affect thermal comfort in different ways even when their values

change by the same amount (e.g., a $1°C$ change in air temperature affects thermal comfort more than a 1% change in air humidity), it is necessary to modify the Euclidean distance for each parameter. To correct for the uneven effects of indoor thermal factors on thermal comfort, this adjustment is made using Eq. (3) [32].

$$H(x_i, x_j) = \left(\sum_{h=1}^{m} \left(w_h * \left|x_i^{(h)} - x_j^{(h)}\right|^2\right)\right)^{\frac{1}{2}} \quad (3)$$

The weight (w_h) allotted to each indoor thermal parameter influencing thermal comfort is calculated using the equation. To find the K data points that show the closest closeness to the test location, distances are calculated. The feedback that was obtained from the individuals at the present test point is then identified as the feedback that occurs the most often out of these K data points. The ideal value for K, which determines the necessary number of data points, may be found with the use of cross-validation. It is crucial to choose a K value that falls in the middle of the two extremes for best results. A low value of K may cause the model to become too sensitive to sample points that are near to the test point, which might lead to an excessive impact from noise points. On the other hand, if K is set very high, the accuracy of the model can suffer [33]. Fig. 2 presents the flowchart illustrating the KNN process.

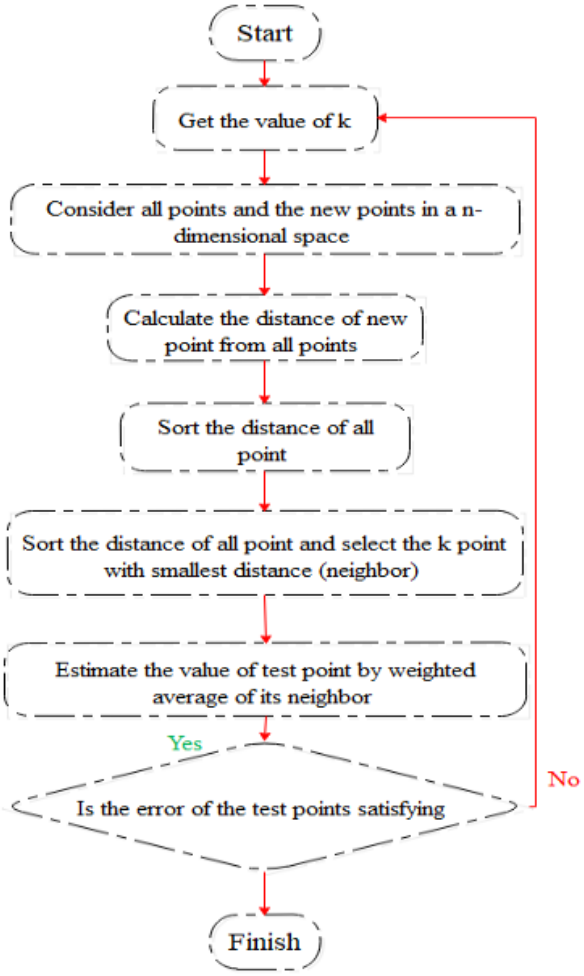


Fig. 2. The flowchart of the KNN model.

C. Dynamic Arithmetic Optimization (DAO)

The core arithmetic optimization algorithm has been improved by introducing a novel accelerator function integrating two dynamic elements to boost performance [34]. In the optimization procedure, the dynamic version adjusts the search phase and candidate solutions by modifying their exploration and exploitation behavior. A standout feature of DAOA is its unique quality of not necessitating any initial parameter fine-tuning, unlike the latest metaheuristic methods [35].

1) A dynamic accelerated function for DAOA: In a dynamic environment, the search phase of the arithmetic optimization algorithm is significantly affected by the DAF. To tailor the AOA for this dynamic context, alterations are required for the accelerated function's initial Min and Max values. However, an ideal scenario would entail an algorithm that isn't reliant on internally adjustable parameters, as an alternative descending function can substitute the DAF [36]. The modification factor within the optimization algorithm is delineated as follows in Eq. (4):

$$DAF = \left(\frac{It_{Max}}{It}\right)^{a} \quad (4)$$

It represents the current iteration count, It_{Max} signifies the maximum allowable number of iterations, and a stands for a constant value. This function diminishes with each successive iteration of the algorithm [37].

2) A dynamic candidate solution for DAOA: The dynamic characteristics of candidate solutions in DAOA are presented in this section. In the case of metaheuristic algorithms, the importance of the exploitation and exploration phases cannot be overstated, and ensuring a proper balance between them is deemed vital for the success of the algorithm. The dynamic iteration of the algorithm seeks to improve both the exploitation and exploration phases by continuously adjusting the position of each solution according to the best solution obtained thus far in the optimization process. In the improved iteration, the Dynamic Candidate Solution (DCS) function is alternatively incorporated into Eq. (5) and Eq. (6) [38].

$$x_{i,j} = (C_{it+1}) = \begin{cases} best(x_j) \div (DCS + \in) \times ((UB_j - LB_j) \times \mu + LB_j)), r2 < 0.5 \\ best(x_j) \times DCS \times ((UB_j - LB_j) \times \mu + LB_j)), Otherwise \end{cases} \quad (5)$$

$$x_{i,j} = (C_{it+1}) = \begin{cases} best(x_j) - DCS \times ((UB_j - LB_j) \times \mu + LB_j)), r3 < 0.5 \\ best(x_j) + DCS \times ((UB_j - LB_j) \times \mu + LB_j)), Otherwise \end{cases} \quad (6)$$

The influence of the decreasing percentage in the candidate solution is considered by introducing the DCS function. Its value is diminished with each iteration of the algorithm, as depicted below in Eq. (7) and Eq. (8):

$$DCS(0) = 1 - \sqrt{\frac{It}{It_{Max}}} \quad (10)$$

$$DCS(t+1) = DCS(t) \times 0.99 \quad (11)$$

The empirical observations gathered from multiple search agents and iterations provide substantial evidence that the integration of candidate solutions in DAOA effectively expedites the convergence rate of AOA [39]. These enhancements result in improved solution quality. Metaheuristic algorithms operating without extensive parameter tuning are typically considered advantageous. This algorithm benefits from adaptive parameters, reducing the parameter tuning requirements to just two variables - maximum iteration and population size. This contrasts competing algorithms, which often necessitate adjustments across various parameters for different problem instances. However, one drawback of this algorithm (see Eq. (9)) lies in its adaptive mechanism, which is based on the iteration count rather than fitness improvement.

$$k_{i,d}^{v} = p_{i,d}^{it} + r_{7,d} \times r_{8,d} \times \left(\left(g_d^{it} + p_{i+1,d}^{it} - 2 \times p_{i,d}^{it}\right) + s_{i,d}^{it+1}\right) \quad (9)$$

D. Wild Geese Algorithm (WGA)

Inspired by the coordinated migratory behavior of wild geese and their patterns of reproduction and death, the Wild Geese Algorithm (WGA) is a metaheuristic algorithm that uses swarm intelligence [40]. The suggested WGA is mainly intended for high-dimensional problem optimization, and it is distinguished by its simplicity and efficacy. The proposed phases of the WGA, to put it broadly, include the following: The wild geese's life cycle, migration, and subsequent evolution are covered in the following sections: a) Velocity displacement and migration stage; b) roaming and searching for food within their native environment; and c) the species' propagation and evolutionary phase [41].

$$k_{i,d}^{w} = p_{i,d}^{it} + r_{9,d} \times r_{10,d} \times (p_{i+1,d}^{it} - p_{i,d}^{it}) \quad (10)$$

First, a population of wild geese is established, and ki is used to represent the positional vector of each wild goose. Next, for every person, the best local location, or personal best solution pi, and the migration velocity Si are ascertained. The target function is then used to rate every wild goose population from best to worst, ranking them in decreasing order [41].

a) Phase of velocity displacement and migration: The wild geese migration is a meticulously organized collective movement characterized by coordination and control. It hinges on the leadership of specific individuals within the sorted population and their neighboring companions to steer the migration. Eq. (10) and Eq. (11) furnish the formulas for velocity and displacement concerning the coordinated velocity of the geese [42].

$$\begin{aligned} s_{i,d}^{It+1} &= \left(r_{1,d} \times s_{i,d}^{It} + r_{2,d} \times \left(s_{i+1,d}^{It} - s_{i-1,d}^{It}\right)\right) + r_{3,d} \\ &\quad \times \left(P_{i,d}^{It} - k_{i-1,d}^{It}\right) \\ &+ r_{4,d} \times \left(P_{i+1,d}^{It} - k_{i,d}^{It}\right) + r_{5,d} \times \left(P_{i+2,d}^{It} - k_{i+1,d}^{It}\right) - r_{6,d} \times \\ &\quad \left(P_{i-1,d}^{It} - k_{i+2}^{It}\right) \end{aligned} \quad (11)$$

Regarding the $i-th$ wild goose, the variables $ki, d, pi, d, and\ si, d$ correspond to the $d-th$ dimension of the current velocity, current position, and best position, respectively. As demonstrated in Eq. (11), the velocities of its nearby members affect the changes in location and velocity of a particular wild goose, such as the $i-th$ wild goose, denoted as $(s_{i+1}^{It} - s_{i-1}^{It})$, along with the positions of neighboring members. The wild geese depend on their neighboring individuals within the sorted population to acquire movement patterns and guidance to minimize the distances between them and these adjacent members.

$$k_{i-1}^{it} \rightarrow p_i^{it}, x_i^{it} \rightarrow p_{i+1}^{it}, k_{i+1}^{it} \rightarrow p_{i+2}^{it}, k_{i+2}^{it} \rightarrow -p_{i-1}^{it}$$

Moreover, the collective movement of the entire flock is directed by the global best member [43]. Eq. (9) depicts this coordinated and sequential positional adjustment, executed in tandem with the leading members, to mimic the motion of all members systematically.

Within Eq. (9), g_d signifies the best position among all members of the group.

b) Roaming about in their native environment and gathering food: The purpose of this step is to incentivize the $i-th$ wild goose to move in the direction of its antecedent, thereby indicating that the $i-th$ wild goose is attempting to approach the $(i+1)-th$ goose $(p_{i+1}^{it} - p_i^{it})$. The formula governing the movement and foraging behavior of the wild goose, denoted as k_i^w, is provided as follows:

c) The process by which wild geese reproduce and evolve: Reproduction and evolution constitute another crucial stage in the life cycle of wild geese. The modeling of reproduction and evolution entails employing a blend of the migration equation (k_i^v) and the movement while searching for food equations (k_i^{wa}), as calculated in Eq. (12). The overall simulations for the proposed WGA algorithm utilize a Cr value of 0.5.

$$k_{i,d}^{it+1} = \begin{cases} k_{i,d}^{v} & \overset{if}{\rightarrow} r_{i,d} \leq cr \\ k_{i,d}^{wa} & \end{cases} \quad (12)$$

d) The decline, movement, and progressive development of wild geese: Previous studies that have been published in the literature show that different optimization methods have different effects on addressing different issues depending on the size of the population and the number of iterations. In some cases, such as those involving the $F2$ and $F3$ functions, the population size of the algorithm is more important and has a greater influence than the number of iterations. However, for some functions, like F7 and F8, the number of iterations in the WGA algorithm is more important and has a greater impact than the population size. In order to arrive at a consensual solution, this is necessary. In order to overcome this difficulty and guarantee a balanced algorithm performance across all test functions, the death phase is created. The procedure starts with $N^{initial}$, the initial maximum population size. The less resilient individuals will be progressively eliminated from the population as the algorithm iterations continue on, according to the standards given in Eq. (13). At the end of the last iteration, the population size will finally reach the final number, $N^{initial}$, after decreasing linearly over time.

$$N = round\left(\frac{N^{initial}}{-((N^{In}-N^{f})*\left(\frac{FV}{FV\,max}\right))}\right) \quad (13)$$

The FV represents the count of function evaluations in Eq. (13).

E. Performance Evaluators

Table II outlines the formulations of various performance metrics used to evaluate the model's accuracy and effectiveness in predicting outcomes. These metrics provide valuable insights into the model's performance:

- Predicted values are represented as b_i.
- Measured values are indicated as m_i.
- The symbol n signifies the sample size.
- The mean of the predictor variable within the dataset is denoted as $\bar{x}$.
- The mean of the measured values is represented as $\bar{m}$, and the mean of the predicted values is denoted as $\bar{b}$.

F. Hyperparameter

Table III lists key hyperparameters for KNWG and KNDA models. In the KNWG model, setting n_neighbors to 01 means just the nearest neighbor is evaluated for predictions. This can provide a more localized prediction strategy. We set leaf_size to 3. This option describes the number of sites where the algorithm transitions from tree-based to brute-force search. Smaller leaf_sizes can improve memory efficiency. To specify the power parameter for the Minkowski distance metric, set the p parameter to 3. A value of 3 represents the Euclidean distance, frequently utilized for its balanced treatment of dimensions. In contrast, the KNDA model uses only the closest neighbor by setting the n_neighbors hyperparameter to 1. While this setting may increase model variance, it may be advantageous in certain situations. Configuring leaf_size to 999 indicates a higher leaf size than KNWG. Selecting this option can improve memory consumption and computational performance, especially for bigger datasets. Setting p to 999 indicates a high power parameter for the Minkowski distance measure. A high number can dramatically impact distance calculation, potentially affecting model behavior in sophisticated ways. Hyperparameter selection should be based on dataset properties and desired objectives, as they greatly affect model behavior and performance. Fine-tuning parameters through testing and validation can improve model performance and generalization across varied datasets and applications.

TABLE II. THE FORMULATIONS OF THE PERFORMANCE METRICS

Coefficient Correlation (R2)	$R^2 = \left(\frac{\sum_{i=1}^{n}(b_i-\bar{b})(m_i-\bar{m})}{\sqrt{\left[\sum_{i=1}^{n}(b_i-\bar{b})^2\right]\left[\sum_{i=1}^{n}(m_i-\bar{m})^2\right]}}\right)^2$	(14)
Root Mean Square Error (RMSE)	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(m_i-b_i)^2}$	(15)
Mean Square Error (MSE)	$\text{MSE} = \frac{1}{n}\sum_{j=1}^{n}(m_i-b_i)^2$	(16)
Prediction Interval (PI)	$PI = \pm t \times SE \times \sqrt{(1+\frac{1}{n}+\frac{(x^*-\bar{x})^2}{\Sigma(x_i-\bar{x})^2})}$	(17)
Mean Absolute Percentage Error (MAPE)	$MAPE = \frac{100}{n}\sum_{i}^{n}\frac{\lvert b_i\rvert}{\lvert m_i\rvert}$	(18)

TABLE III. THE HYPERPARAMETER FOR MODELS

Models	**Hyperparameter**		
	n_neighbors	***leaf_size***	***p***
KNWG	01	3	3
KNDA	1	999	999

III. RESULT AND DISCUSSION

Table IV presents the performance metrics for the developed models in the context of KNN. These models were evaluated across different phases: Training, Validation, Test, and All (combining all data). The performance metrics include RMSE, R^2, MSE, PI, and MAPE.

1) KNN model: The KNN model demonstrates strong predictive capabilities. In the training phase, it achieves an RMSE of 1.525 and an R^2 of 0.975, indicating a high level of accuracy. Similar results are observed in the validation and test phases, with slight increases in RMSE and a decrease in R^2, which is expected as the model generalizes to new data. When considering all data, the KNN model maintains a solid performance.

2) KNWG model: The KNWG model outperforms the KNN model across all phases. It exhibits significantly lower RMSE values, indicating improved accuracy. In the training phase, it achieves an impressive RMSE of 0.680 and a high R^2 of 0.995, highlighting its superior performance. This trend continues in the validation and test phases. When considering

all data, the KNWG model consistently maintains lower RMSE and higher R^2 values compared to the KNN model.

3) KNDA model: The KNDA model, while not as accurate as the KNWG model, still demonstrates respectable performance. It achieves RMSE values higher than KNWG but lower than the KNN model across all phases. In the training phase, it records an RMSE of 1.078 and an R^2 of 0.987. Similar trends are observed in the validation and test phases. When considering all data, the KNDA model offers a balanced performance.

Overall, the results indicate that the KNWG model is the most accurate among the three, followed by KNDA, with KNN being the least accurate. The choice of the best model depends on the specific application's requirements. Additionally, these models exhibit low MAPE values across all phases, confirming their reliability. These findings provide valuable insights into selecting an appropriate model for CL prediction and its potential applications in various domains.

Table V compares the performance metrics of the presented study with those of published articles. In terms of RMSE, Moradzadeh et al. achieved 0.4832, while Roy et al. achieved the lowest RMSE of 0.059. Gong et al. and Afzal et al. obtained RMSE values of 0.1929 and 1.4122, respectively. Regarding the R2 values, Moradzadeh et al. recorded the highest at 0.9993, followed closely by the present study at 0.996. Roy et al. achieved an R2 of 0.99, while Gong et al. and Afzal et al. attained R2 values of 0.9882 and 0.9806, respectively. These comparisons provide insights into the relative performance of the presented study in relation to existing research in the field.

Fig. 3 provides a visual representation highlighting the differences among R^2, RMSE, and MSE for the proposed models. It is evident from the graph that the KNWG model stands out as the top performer, showcasing the lowest RMSE and MSE values, signifying its outstanding predictive accuracy in estimating CL. Furthermore, it attains the highest R^2 values among the models, underscoring its robust performance. Moreover, the Fig. 3 diagram emphasizes the intermediate performance of the KNAO model. It displays a well-balanced performance, occupying a middle position between the precision achieved by the KNWG model and the outcomes of the KNN model. Conversely, the KNN model, functioning as an independent model, displayed the least accurate results in comparison to the other models.

Fig. 4 displays a scatter plot that illustrates the performance of the models concerning their R^2 and RMSE values. The plot distinguishes each model's three phases—train, validation, and test—using unique circular markers in different colors. These markers cluster around a central line, symbolizing the ideal R^2 value 1, signifying a perfect match between predicted and actual values. A more in-depth examination of the data points linked to the KNWG model within the plot uncovers a tight cluster near the central line. This clustering stands as evidence of the model's precision in predicting values, as it consistently maintains proximity to the ideal R^2 value.

TABLE IV. THE RESULT OF DEVELOPED MODELS FOR KNN

Model	phase	Index values				
		RMSE	***R2***	***MSE***	***PI***	***MAPE***
KNN	Train	1.525	0.975	2.326	0.031	3.878
	Validation	1.871	0.968	3.500	0.039	4.363
	Test	1.944	0.963	3.777	0.040	5.176
	All	1.649	0.971	2.719	0.034	4.145
KNWG	Train	0.680	0.995	0.463	0.014	2.703
	Validation	1.020	0.990	1.040	0.021	3.135
	Test	1.388	0.980	1.927	0.028	3.829
	All	0.877	0.991	0.769	0.018	2.936
KNDA	Train	1.078	0.987	1.163	0.022	4.386
	Validation	1.666	0.978	2.775	0.034	3.600
	Test	1.824	0.972	3.326	0.037	3.637
	All	1.315	0.981	1.728	0.027	4.156

TABLE V. COMPARISON BETWEEN THE PRESENTED AND PUBLISHED ARTICLES

Articles	Index values	
	RMSE	***R^2***
Moradzadeh et al. [44]	0.4832	0.9993
Roy et al. [45]	0.059	0.99
Gong et al. [46]	0.1929	0.9882
Afzal et al. [47]	1.4122	0.9806
Present Study	0.698	0.996

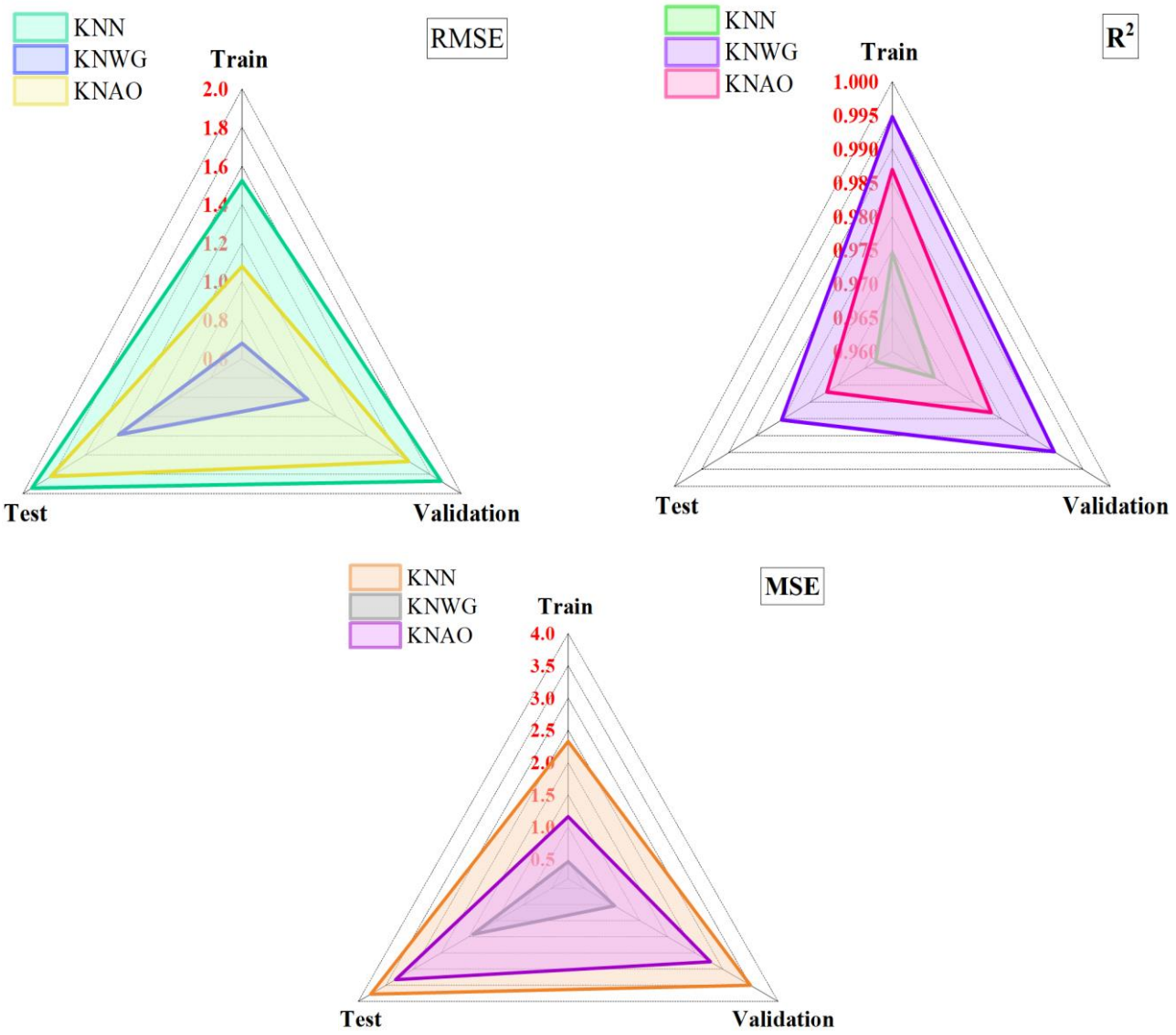


Fig. 3. The comparison of parameters.

In contrast, both the KNDA and KNN models exhibit scattered data points, indicating a wider range of values. This scattering suggests that these models display variations in their predictions and may not consistently achieve high R^2 values. The scatter plot in Fig. 4 underscores the superior predictive accuracy of the KNWG model while highlighting the broader variability in the predictions made by the KNDA and KNN models.

Fig. 5 provides a symbolic representation that visually conveys the error percentages associated with each model. Analyzing model errors is a vital method to evaluate their precision. This plot assists in evaluating the models' performance across the training, testing, and validation phases. It's worth highlighting that the KNN model displays a higher error rate compared to the other models, with the maximum recorded error percentage reaching 20%. In contrast, KNWG has demonstrated the utmost accuracy among all the models. In the testing phase, the highest observed error for KNWG is 10%, and a substantial portion of its data points cluster closely around a minimal 0% error. Meanwhile, the KNDA model exhibits moderate performance, with the highest error percentage reaching 15% in the testing phase. It consistently maintains moderate error values when compared to the other models.

Fig. 6 portrays the distribution patterns of the proposed models using a violin plot, which represents the 3 stages of train, validation, and test. It's noticeable that the data points for the KNN model display a broad dispersion, covering error percentages spanning from 20 to -20, which is particularly pronounced during the training phase. To efficiently detect outlier data points for model comparison, a range equal to 1.5 times the Interquartile Range (IQR) is utilized. In contrast, KNWG's data points are closely grouped within the error percentage range of 10 to -10, while KNDA data points fall within the range of 15 to -15 percent error.

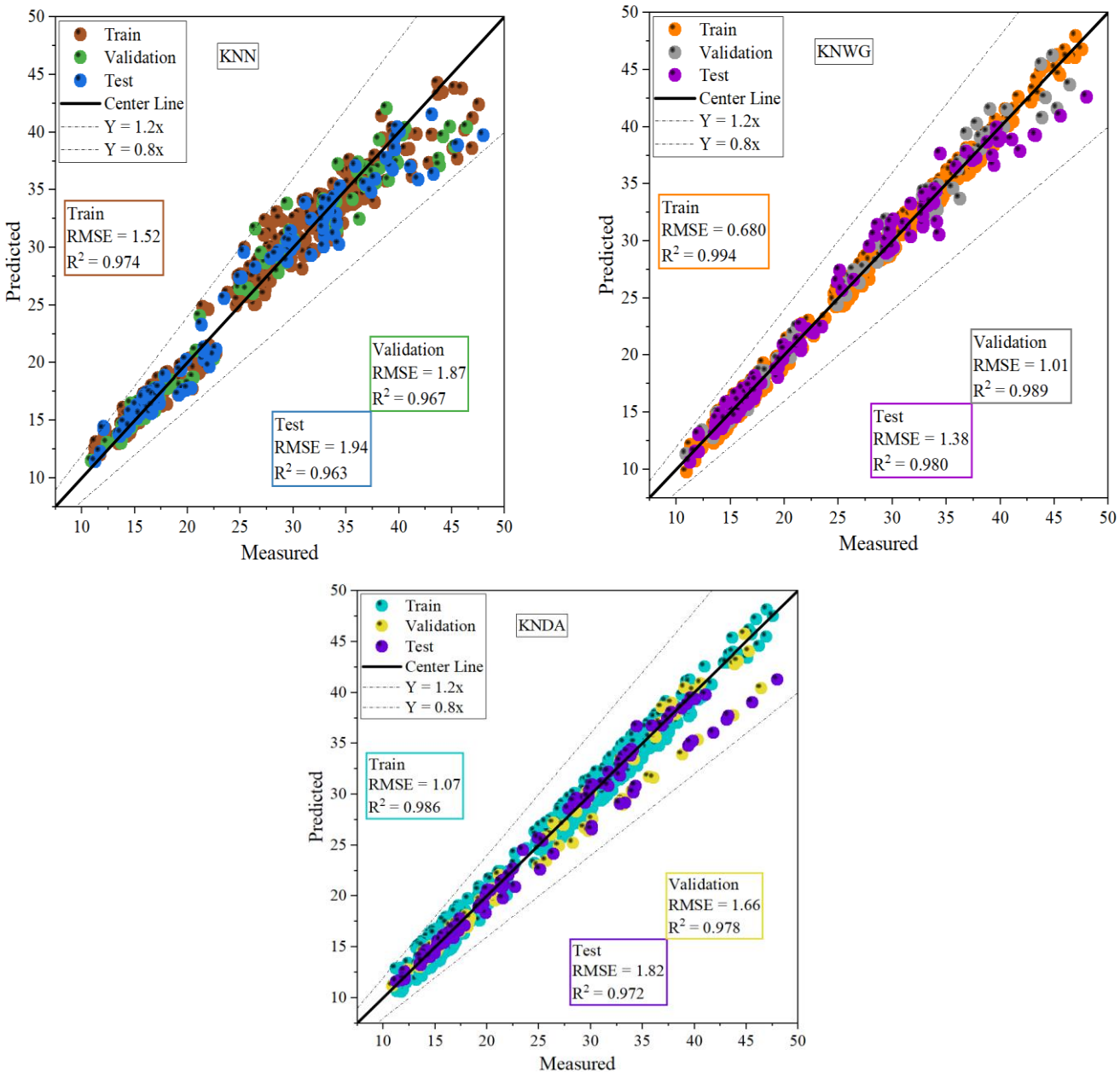
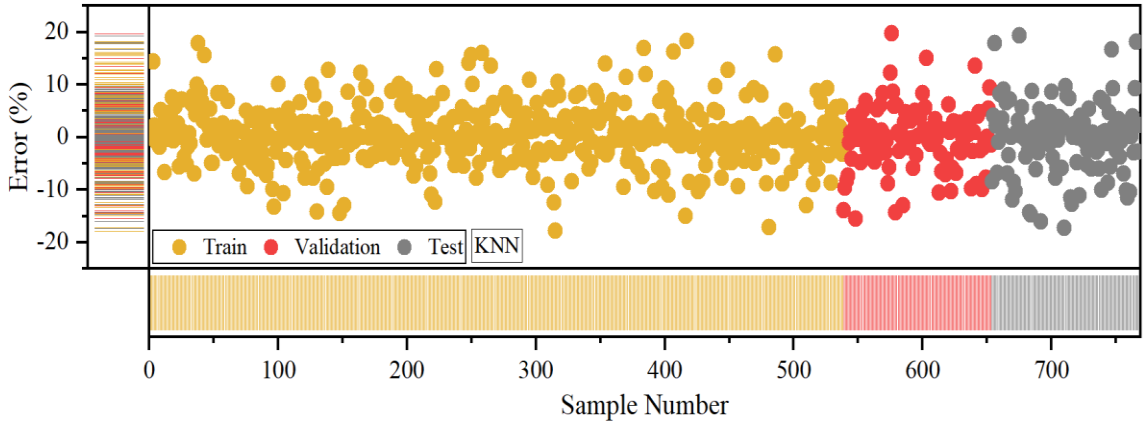


Fig. 4. The scatter plot for developed models.



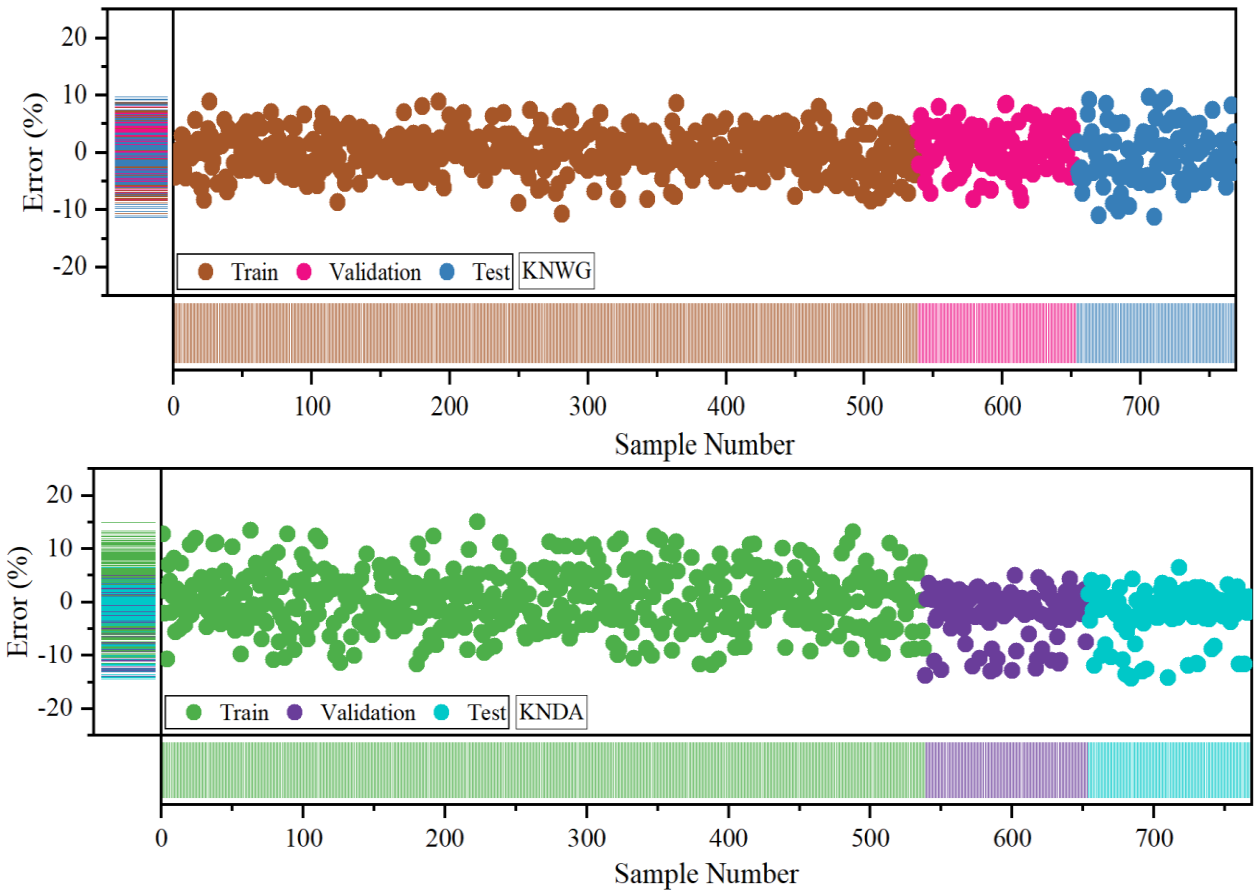


Fig. 5. The error percentage for the models is based on the symbol plot.

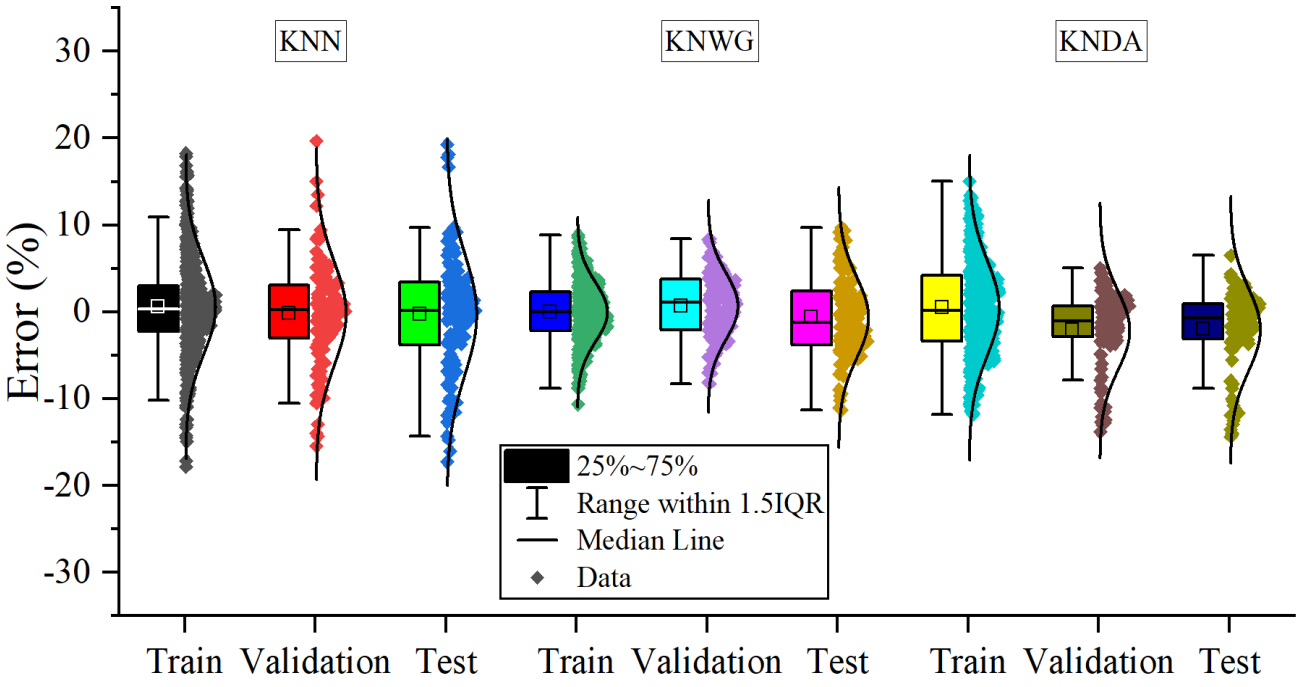


Fig. 6. The box of errors among the developed models.

IV. CONCLUSION

This article delves into developing and evaluating predictive models, specifically focusing on K-nearest neighbors (KNN) for estimating Cooling Load (CL) in buildings. Three distinct models were examined: KNN, KNWG, and KNDA. The study encompassed various phases, including training, validation, and testing, providing a comprehensive analysis of their performance. The results and discussions highlight the superiority of the KNWG model, which consistently demonstrated exceptional predictive accuracy with the lowest Root Mean Square Error (RMSE) and Mean Square Error (MSE) values. Its high Coefficient Correlation (R^2) values emphasize its robust overall performance.

On the other hand, while functional, the KNN model exhibited less accuracy and higher error rates than the other models. The error analysis further solidifies the KNWG model's precision, with most data points clustering closely around minimal error percentages. KNDA, while not as accurate as KNWG, maintained moderate error values consistently across phases. The distribution patterns and outlier detection methods provided additional insights into the models' performance. KNWG and KNDA exhibited narrower error ranges, indicating their stability and reliability. The KNWG model is the most accurate and reliable option for predicting building CL. Its consistently superior performance in multiple phases and various evaluation metrics makes it a valuable tool for building energy efficiency applications. This research contributes to the advancement of predictive modeling techniques and their potential for real-world applications in improving energy efficiency in residential buildings.

Despite its advancements, this study has limitations worth noting. Firstly, the proposed approach heavily relies on historical data, potentially limiting its applicability to new or unique building designs or environments. Secondly, while KNN, DAO, and WGA are powerful techniques, their performance may vary depending on specific datasets and configurations, necessitating careful tuning. Additionally, the study's focus on residential buildings may not fully capture the complexities of larger commercial or industrial structures. Moreover, the integration of KNN with DAO and WGA introduces additional computational complexities, potentially hindering real-time application in some scenarios. Lastly, external factors such as climate change could impact the model's long-term accuracy.

REFERENCES

[1] Q. Zhang, Z. Tian, Z. Ma, G. Li, Y. Lu, and J. Niu, "Development of the heating load prediction model for the residential building of district heating based on model calibration," Energy, vol. 205, p. 117949, 2020.

[2] Y. Zhang, Z. Zhou, J. Liu, and J. Yuan, "Data augmentation for improving heating load prediction of heating substation based on TimeGAN," Energy, vol. 260, p. 124919, 2022.

[3] S. S. Roy, P. Samui, I. Nagtode, H. Jain, V. Shivaramakrishnan, and B. Mohammadi-Ivatloo, "Forecasting heating and cooling loads of buildings: A comparative performance analysis," J Ambient Intell Humaniz Comput, vol. 11, pp. 1253–1264, 2020.

[4] S. Shamshirband et al., "Heat load prediction in district heating systems with adaptive neuro-fuzzy method," Renewable and Sustainable Energy Reviews, vol. 48, pp. 760–767, 2015.

[5] K. Kato, M. Sakawa, K. Ishimaru, S. Ushiro, and T. Shibano, "Heat load prediction through recurrent neural network in district heating and cooling systems," in 2008 IEEE international conference on systems, man and cybernetics, IEEE, 2008, pp. 1401–1406.

[6] J. Ling, N. Dai, J. Xing, and H. Tong, "An improved input variable selection method of the data-driven model for building heating load prediction," Journal of Building Engineering, vol. 44, p. 103255, 2021.

[7] G. Xue, Y. Pan, T. Lin, J. Song, C. Qi, and Z. Wang, "District heating load prediction algorithm based on feature fusion LSTM model," Energies (Basel), vol. 12, no. 11, p. 2122, 2019.

[8] M. Protić et al., "Appraisal of soft computing methods for short term consumers' heat load prediction in district heating systems," Energy, vol. 82, pp. 697–704, 2015.

[9] E. Guelpa, L. Marincioni, M. Capone, S. Deputato, and V. Verda, "Thermal load prediction in district heating systems," Energy, vol. 176, pp. 693–703, 2019.

[10] M. Sajjad et al., "Towards efficient building designing: Heating and cooling load prediction via multi-output model," Sensors, vol. 20, no. 22, p. 6419, 2020.

[11] A. Moradzadeh, A. Mansour-Saatloo, B. Mohammadi-Ivatloo, and A. Anvari-Moghaddam, "Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings," Applied Sciences, vol. 10, no. 11, p. 3829, 2020.

[12] G. Xue, C. Qi, H. Li, X. Kong, and J. Song, "Heating load prediction based on attention long short term memory: A case study of Xingtai," Energy, vol. 203, p. 117846, 2020.

[13] C. Wang et al., "Research on thermal load prediction of district heating station based on transfer learning," Energy, vol. 239, p. 122309, 2022.

[14] Y. Ding, Q. Zhang, T. Yuan, and K. Yang, "Model input selection for building heating load prediction: A case study for an office building in Tianjin," Energy Build, vol. 159, pp. 254–270, 2018.

[15] F. Dalipi, S. Yildirim Yayilgan, and A. Gebremedhin, "Data-driven machine-learning model in district heating system for heat load prediction: A comparison study," Applied Computational Intelligence and Soft Computing, vol. 2016, 2016.

[16] R. Chaganti et al., "Building heating and cooling load prediction using ensemble machine learning model," Sensors, vol. 22, no. 19, p. 7692, 2022.

[17] H. Khajavi and A. Rastgoo, "Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms," Energy, vol. 272, p. 127069, 2023.

[18] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," Energy, vol. 182, pp. 72–81, 2019.

[19] A. Moradzadeh, A. Mansour-Saatloo, B. Mohammadi-Ivatloo, and A. Anvari-Moghaddam, "Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings," Applied Sciences, vol. 10, no. 11, p. 3829, 2020.

[20] P. Penna, A. Prada, F. Cappelletti, and A. Gasparella, "Multi-objectives optimization of Energy Efficiency Measures in existing buildings," Energy Build, vol. 95, pp. 57–69, 2015, doi: https://doi.org/10.1016/j.enbuild.2014.11.003.

[21] L. T. Le, H. Nguyen, J. Dou, and J. Zhou, "A comparative study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in estimating the heating load of buildings' energy efficiency for smart city planning," Applied Sciences, vol. 9, no. 13, p. 2630, 2019.

[22] S. S. K. Kwok and E. W. M. Lee, "A study of the importance of occupancy to building cooling load in prediction by intelligent approach," Energy Convers Manag, vol. 52, no. 7, pp. 2555–2564, 2011.

[23] C. Fan and Y. Ding, "Cooling load prediction and optimal operation of HVAC systems using a multiple nonlinear regression model," Energy Build, vol. 197, pp. 7–17, 2019, doi: https://doi.org/10.1016/j.enbuild.2019.05.043.

[24] T. Chaudhuri, Y. C. Soh, H. Li, and L. Xie, "A feedforward neural network based indoor-climate control framework for thermal comfort and energy saving in buildings," Appl Energy, vol. 248, pp. 44–53, 2019, doi: https://doi.org/10.1016/j.apenergy.2019.04.065.

[25] Z. Yu, F. Haghighat, B. C. M. Fung, and H. Yoshino, "A decision tree method for building energy demand modeling," Energy Build, vol. 42, no. 10, pp. 1637–1646, 2010, doi: https://doi.org/10.1016/j.enbuild.2010.04.006.

[26] S. S. Roy, P. Samui, I. Nagtode, H. Jain, V. Shivaramakrishnan, and B. Mohammadi-Ivatloo, "Forecasting heating and cooling loads of buildings: A comparative performance analysis," J Ambient Intell Humaniz Comput, vol. 11, pp. 1253–1264, 2020.

[27] J.-S. Chou and D.-K. Bui, "Modeling heating and cooling loads by artificial intelligence for energy-efficient building design," Energy Build, vol. 82, pp. 437–446, 2014, doi: https://doi.org/10.1016/j.enbuild.2014.07.036.

[28] D. Lixing, L. Jinhu, L. Xuemei, and L. Lanlan, "Support vector regression and ant colony optimization for HVAC cooling load prediction," in 2010 international symposium on computer, communication, control and automation (3ca), IEEE, 2010, pp. 537–541.

[29] A. Moradzadeh, A. Mansour-Saatloo, B. Mohammadi-Ivatloo, and A. Anvari-Moghaddam, "Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings," Applied Sciences, vol. 10, no. 11, p. 3829, 2020.

[30] S. Afzal, B. M. Ziapour, A. Shokri, H. Shakibi, and B. Sobhani, "Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms," Energy, vol. 282, p. 128446, 2023.

[31] L. Xiong and Y. Yao, "Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm," Build Environ, vol. 202, p. 108026, 2021.

[32] H. A. Abu Alfeilat et al., "Effects of distance measure choice on k-nearest neighbor classifier performance: a review," Big Data, vol. 7, no. 4, pp. 221–248, 2019.

[33] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," Sci Rep, vol. 12, no. 1, p. 6256, 2022.

[34] N. Khodadadi, V. Snasel, and S. Mirjalili, "Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints," IEEE Access, vol. 10, pp. 16188–16208, 2022.

[35] İ. Gölcük, F. B. Ozsoydan, and E. D. Durmaz, "An improved arithmetic optimization algorithm for training feedforward neural networks under dynamic environments," Knowl Based Syst, vol. 263, p. 110274, 2023.

[36] V. Panneerselvam and R. Thiagarajan, "ACBiGRU-DAO: Attention Convolutional Bidirectional Gated Recurrent Unit-based Dynamic Arithmetic Optimization for Air Quality Prediction," Environmental Science and Pollution Research, vol. 30, no. 37, pp. 86804–86820, 2023.

[37] H. Weirong, T. A. N. Pengcheng, W. Shuqing, and P. Li, "A comparison of arithmetic operations for dynamic process optimization approach," Chin J Chem Eng, vol. 18, no. 1, pp. 80–85, 2010.

[38] R. Thota and N. Sinha, "An enhanced arithmetic optimization algorithm for global maximum power point tracking of photovoltaic systems under dynamic irradiance patterns," Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 44, no. 4, pp. 10116–10134, 2022.

[39] M. von Andrian and R. D. Braatz, "Stochastic dynamic optimization and model predictive control based on polynomial chaos theory and symbolic arithmetic," in 2020 American Control Conference (ACC), IEEE, 2020, pp. 3399–3404.

[40] M. Ghasemi, A. Rahimnejad, R. Hemmati, E. Akbari, and S. A. Gadsden, "Wild Geese Algorithm: A novel algorithm for large scale optimization based on the natural life and death of wild geese," Array, vol. 11, p. 100074, 2021.

[41] T. T. Nguyen, T. L. Duong, and T. Q. Ngo, "Wild geese algorithm for the combination problem of network reconfiguration and distributed generation placement," International Journal on Electrical Engineering and Informatics, vol. 14, no. 1, pp. 76–91, 2022.

[42] B. Deepanraj, N. Senthilkumar, T. Jarin, A. E. Gurel, L. S. Sundar, and A. V. Anand, "Intelligent wild geese algorithm with deep learning driven short term load forecasting for sustainable energy management in microgrids," Sustainable Computing: Informatics and Systems, vol. 36, p. 100813, 2022.

[43] T. T. Nguyen, T. T. Nguyen, and T. D. Nguyen, "Minimizing electricity cost by optimal location and power of battery energy storage system using wild geese algorithm," Bulletin of Electrical Engineering and Informatics, vol. 12, no. 3, pp. 1276–1284, 2023.

[44] A. Moradzadeh, A. Mansour-Saatloo, B. Mohammadi-Ivatloo, and A. Anvari-Moghaddam, "Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings," Applied Sciences, vol. 10, no. 11, p. 3829, 2020.

[45] S. S. Roy, P. Samui, I. Nagtode, H. Jain, V. Shivaramakrishnan, and B. Mohammadi-Ivatloo, "Forecasting heating and cooling loads of buildings: A comparative performance analysis," J Ambient Intell Humaniz Comput, vol. 11, pp. 1253–1264, 2020.

[46] M. Gong, Y. Bai, J. Qin, J. Wang, P. Yang, and S. Wang, "Gradient boosting machine for predicting return temperature of district heating system: A case study for residential buildings in Tianjin," Journal of Building Engineering, vol. 27, p. 100950, 2020.

[47] S. Afzal, B. M. Ziapour, A. Shokri, H. Shakibi, and B. Sobhani, "Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms," Energy, p. 128446, Jul. 2023, doi: 10.1016/j.energy.2023.128446.