Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.754966

HYBRID ARTIFICIAL INTELLIGENCE APPROACH (SVR-PSO) FOR MODELLING THE DYE REMOVAL

Amel Hebboul[1*], Ouafa Tobbi[2], Hichem Fiala[3], Zhour Hattab[3]
[1*]Ecole Normale Supérieure El Katiba Assia Djebar, 25000, Constantine 3, Algeria, ahebboul@yahoo.fr
[2] Department of science and technology, faculty of technology, Universty of Mostapha Benboulaid Batna 2, Batna, Algeria
[3] Laboratory of treatment of wastewater and valorization of industrial waste, departement of chemistry, Annaba, Algeria
***Corresponding author: -** Amel Hebboul
[*]Ecole Normale Supérieure El Katiba Assia Djebar, 25000, Constantine 3, Algeria, ahebboul@yahoo.fr

ABSTRACT

Artificial intelligence (AI) has gathered significant attention and acceptance across various domains, including water treatment and desalination, where it has proven to be a valuable tool for enhancing process efficiency and addressing water pollution and scarcity. AI techniques offer optimized chemical usage, reduced operational costs, and effective solutions. In this paper, a hybrid method, SVR-PSO is proposed for modeling the dye removal process. First, Particle Swarm Optimization (PSO) tunes the best hyperparameters for Support Vector Regression (SVR). Subsequently, Support Vector Regression (SVR) is built using the obtained best hyperparameters and then fitted. The proposed method is tested and evaluated for modeling the removal of The Malachite Green dye using Helianthus Annuus seed shells as an eco-friendly adsorbent. The proposed method is compared with SVR tuned with Genetic Algorithm (SVR-GA) and SVR tuned with the Grid Search method (SVR-GS), and the obtained results demonstrate the efficiency of the proposed method.

Keywords: Support Vector Regression, Particle Swarm Optimization, Genetic Algorithms, dye removal.

1 Introduction

Water, the essence of life, encompasses 71% of the Earth's surface and constitutes over half of the human body[1], [2]. However, the widespread use of dyes across various industries presents a formidable threat to water ecosystems due to their toxicity, poor biodegradability, and vivid coloration[3]. Industrial wastewater containing these colour compounds poses a dire risk to ecosystems by disrupting photosynthesis and directly harming living organisms, thereby jeopardizing water quality and human health[4]. To mitigate these detrimental effects, the removal of dyes from water is imperative. Various methods including physical, chemical, and biological treatments have been employed to achieve this goal. Physical techniques entail the separation of dye particles through processes such as filtration, adsorption, and membrane filtration[5], [6]. Chemical methods involve reactions such as oxidation, coagulation, or precipitation to transform or eliminate dyes, while biological methods utilize organisms to

biodegrade or metabolize dye compounds[6]. Among these techniques, adsorption, utilizing adsorbent materials, stands out for its efficacy and versatility in water treatment[7], [8], [9].

The application of Artificial Intelligence (AI) technologies has revolutionized numerous fields by mimicking human-like capabilities such as language interpretation, image recognition, problem-solving, and data-driven learning[10]. AI, categorized into machine learning, deep learning, and data analytics, has found extensive applications in intelligent decision-making, blockchain, cloud computing, the Internet of Things (IoT), and the fourth industrial revolution (Industry 4.0)[11]. In recent years, AI has emerged as a promising approach to enhance traditional methods of dye removal[12], [13]. AI technologies offer the potential to optimize various aspects of dye removal processes, including reducing experimental costs, optimizing chemical usage, improving efficiency, and enhancing overall performance[13].

The commonly employed AI methods in water treatment include Artificial Neural Networks (ANN) and its variants such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Feed-Forward Backpropagation Neural Network (FFBPNN), and Adaptive Network-Based Fuzzy Inference System (ANFIS)[12]. Additionally, Decision Trees (DT) and Support Vector Regression (SVR) have gained prominence. Hybrid approaches such as ANN-GA, ANN-PSO, and SVR-GA have also been explored extensively in water treatment research [13].

Support Vector Regression (SVR) is a powerful machine learning algorithm used for regression tasks[14]. It is a variant of the Support Vector Machine (SVM) algorithm[15], which is primarily used for classification. SVR, like SVM, operates by finding the hyperplane that best separates data into different classes. However, instead of focusing on classification boundaries, SVR focuses on fitting the best possible line within a predefined margin of error [16]. What distinguishes SVR from traditional regression methods is its ability to handle complex data distributions and high-dimensional feature spaces effectively. It achieves this by transforming the input data into a higher-dimensional space through a kernel function, where it can find a linear relationship that might not be apparent in the original feature space. This transformation allows SVR to capture nonlinear relationships between variables. Notably, SVR models demonstrate resilience against data volume constraints, providing reliable predictions even with limited datasets. Additionally, SVR is robust to outliers, as it focuses on minimizing errors within the specified margin rather than being heavily influenced by individual data points. However, effective parameter optimization remains a challenge to achieve heightened prediction accuracy[17]. Techniques such as Grid Search (GS) methodology with cross-validation[18], particle swarm optimization (PSO)[19], [20], genetic algorithm (GA)[21], [22], and differential evolution (DE)[23] are commonly employed for this purpose. The robust optimization of SVR model parameters directly influences prediction accuracy and the generalization capability of the model.

SVR has been applied in modelling dye removal processes[13], [24], [25], [26], [27], [28], [29], [26], [30], with various studies employing different optimization techniques for SVR hyperparameter tuning. While some studies utilize the trial-and-error method [24], [26], others employ the grid research method [25], [27], [28], or genetic algorithms [29], [31], [30]. The trial-and-error method and the grid research method are very time-consuming[17].

In this paper, we proposed exploiting Particle Swarm Optimization (PSO) for fine-tuning SVR hyperparameters. We favour PSO over Genetic Algorithm (GA) due to its simpler parameter

setup and superior speed, particularly in continued non-binary space research contexts. This selection of PSO aims to capitalize on its efficiency and effectiveness in optimizing SVR parameters for improved modelling accuracy. To evaluate our proposed method, we conducted a comparative study to model the process of removing Malachite Green (MG) dye using an eco-friendly adsorbent, namely Helianthus Annuus Seeds shells. The experimental study of Malachite Green (MG) dye removal is performed in the Laboratory of water treatment and valorisation of industrial wastes (LTEVDI)[32].

The remainder of the paper is structured as follows: the first section presents the methods used in the proposed approach, such as Support Vector Regression (SVR), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA). In the second section, the proposed method is detailed. The third section demonstrates the efficiency of our proposed method with a comparative study. Finally, we conclude the paper in the last section.

2 Methods

2.1 Support Vector Regression

SVM as a learning method was developed by Vapnik [15] and is a powerful tool. This supervised learning method can be used for regression or classification in nonlinear models, and density estimation leads to complex optimization problems, typically quadratic programming. However, this method (SVM) is often time consuming and difficult to adapt, suffering from the problem of a large memory requirement and CPU time when trained in batch mode. This limitation is overcome by LS-SVM as the modified version of SVM which solves the set of linear equations instead of the quadratic programming problem to minimize the complex nature of the optimization processes The theory and more details of SVM and LS-SVM can be found in the literature [33].

In SVR, the objective is to find a function that approximates the mapping from input variables to continuous output variables, while also minimizing the margin of error, known as epsilon-insensitive tube. This tube represents the range within which errors are acceptable, and data points outside this tube contribute to the error function. SVR aims to find the hyperplane that has the maximum margin within this tube[16].

Given a set of input–output sample pairs $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, the objective of SVR technique is to approximate the nonlinear relationship given in (1), such that f(x) should be as close as possible to the target value y and should be as flat as possible in order to avoid over-fitting.

$$y_i = f(x_i) = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) K(x, x_i) + b \qquad (1)$$

where b is bias value, α_i and α^* are the Lagrange multipliers, $K(x, x_i)$ is the kernel function. To evaluate the goodness of the regression function, the ε -insensitive loss function is used [15]:

$$l = |y_i - f(x_i)|_\varepsilon = \begin{cases} 0, & |y_i - f(x_i)| \le \varepsilon \\ |y_i - f(x_i)| - \varepsilon, | & otherwise \end{cases} \qquad (2)$$

To find the coefficients α_i^* and α_i one must solve the following quadratic optimization problem: maximize the functional:

$$W(\alpha^*,\alpha) = -\varepsilon\sum_{i=1}^{n}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{n} y(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^* - \alpha_i)\left(\alpha_j^* - \alpha_j\right)K\left(x_i, x_j\right) \quad (3)$$

subject to constraints

$$\sum_{i=1}^{n}(\alpha_i^* - \alpha_i) = 0, \quad 0 \le \alpha_i, \alpha_i^* \le C, i = 1, \dots, n$$

Where C is the regularization parameter, determining the trade-off between the fitting error minimization and the smoothness of the estimated function. Many kernel functions have been presented in the literature, such as the linear kernel, radial basis function (RBF), sigmoid kernel and polynomial kernel. The selection of the proper kernel function to map the nonlinear input space into a linear feature space depends on the distribution of the training data in the feature space[33]. The function RBF is broadly employed in regression problems because RBF can be substantially faster to train than other kernel functions. The RBF can be expressed as:

$$k\left(x_i, x_j\right) = \exp\left(-\gamma\left\|x_i - x_j\right\|^2\right) \quad (4)$$

2.2 Genetic Algorithm

Genetic algorithms (GAs) are powerful optimization techniques inspired by the principles of natural selection and genetics[21]. They belong to the broader class of evolutionary algorithms and are particularly adept at solving complex, multi-dimensional optimization problems across various domains. In GA, a population of candidate solutions evolves over generations through the iterative application of selection, crossover, and mutation operations, mimicking the process of natural evolution. Each individual in the population represents a potential solution to the optimization problem, and their fitness is evaluated based on a predefined objective function. Through the probabilistic combination of genetic material from parent individuals and occasional random alterations via mutation, GAs efficiently explore the search space to discover high-quality solutions. Despite their stochastic nature, GAs exhibit robustness and scalability, making them applicable to a wide range of real-world problems, including optimization, machine learning, and engineering design. As highlighted by Holland [34], GAs offer a unique approach to optimization, drawing inspiration from biological evolution to tackle complex optimization challenges.

In a genetic algorithm (GA), there are three fundamental operations: selection, crossover, and mutation. These operations are essential for evolving populations toward optimal solutions. Selection involves choosing individuals based on their fitness, with strategies like roulette wheel, tournament, or rank-based selection. Crossover combines genetic information from parents to create offspring, with techniques like single-point, multi-point, or uniform crossover. Mutation introduces random changes to individual chromosomes, such as bit flips, swaps, or inversions. Various strategies within each operation offer flexibility and adaptability to different problem domains and optimization requirements, making GAs a versatile and powerful tool for evolutionary optimization.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a metaheuristic optimization algorithm inspired by the social behaviour of bird flocking or fish schooling and proposed in 1995 by Russel Eberhart

and James Kennedy[19]. It's commonly used to solve optimization problems, particularly in the domains of engineering, machine learning, and artificial intelligence. In PSO, a population of candidate solutions, called particles, moves through the search space to find the optimal solution. Each particle in the swarm has a velocity v_i^t and a position χ_i^t. The velocity determines the direction and magnitude of the particle's movement, while the position represents a potential solution in the search space. Initially, both velocity and position are randomly initialized. The fitness of each particle is evaluated using the objective function. This function determines how good a particular solution is with respect to the optimization criteria. Each particle remembers its best position (p_i^t) found so far, called personal best position. Among all the personal best positions, the global best position (p_g^t) is the best solution found by any particle in the entire population. The position of each particle is updated based on its current velocity, considering its personal best and the global best positions. The velocity update formula typically considers the inertia of the particle's movement, cognitive component (related to its personal best), and social component (related to the global best).

$$v_i^{t+1} = \omega^t v_i^t + c_1 r_1 (p_i^t - \chi_i^t) + c_2 r_2 (p_g^t - \chi_i^t) \qquad (5)$$

where p_i^t is the best position of the particle i at instant t. p_g^t is the global best position achieved by the swarm at instant t. c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random numbers between 0 and 1. In eq. (5), $\omega^t v_i^t$ is the previous velocity of the particle, $c_1 r_1 (p_i^t - \chi_i^t)$ represents the individual reflation of the particle itself (cognitive component) and $c_2 r_2 (p_g^t - \chi_i^t)$represents the collaboration in the swam (social component). The position of each particle is updated by:

$$\chi_i^{t+1} = \chi_i^t + v_i^{t+1} \qquad (6)$$

3 Proposed method

The proposed method consists of three steps: (1) data collection and normalization, (2) SVR parameters tuning, (3) modelling of the dye removal process.

In the normalization data step, the experimental data is rescaled within a uniform range between 0.1 and 0.9 using the following equation:

$$X = 0.1 + \frac{x - \min(x)}{\max(x) - \min(x)} * 0.8 \qquad (7)$$

Where $\boldsymbol{x}$ is the original value, $\mathbf{min}(\boldsymbol{x})$ and $\mathbf{max}(\boldsymbol{x})$ are the minimum and maximum values in the original range, and $\boldsymbol{X}$ is the rescaled value. The normalized data are split randomly in training and test data set.

In the modelling step, the dye removal process is modelled using the SVR method. First, the hyperparameters of SVR are tuned using the meta-heuristic algorithm PSO. Subsequently, the SVR model created with the best hyperparameters is fitted and tested.

3.1 SVR-PSO

In this study, we have selected Support Vector Regression (SVR) with a Radial Basis Function (RBF) kernel as our model. The RBF function is widely employed in regression problems because it is often faster to train compared to other kernel functions [13], [24], [25], [26], [27], [28], [29], [26]. Training SVR with the RBF kernel requires specifying three parameters: the

regularization parameter (C), the epsilon (ε) SVR model, and the kernel coefficient (gamma γ) for the RBF kernel. The position of the particle is represented by vector of three real numbers. The first element represents the C parameter, the second element represents the (ε) epsilon parameter, and the last one represents the (γ) gamma parameter. The algorithm for the proposed method is **SVR-PSO Algorithm.**

SVR-PSO Algorithm

Input: DB, PSO parameters (c_1, c_2, w, Np, lb, ub)

Output: best SVR parameters ($C, epsilon, gamma$)

Begin

Initializing step:

For all particle i **do**

- Random uniform Initialization of the position χ_i and the velocity v_i within space search limitations.
- Compute the value of the fitness function F_i;
- Set the best personal position and its fitness value:
- $Fp_i = F_i$ and $p_i = \chi_i$

End for

Set the index of best global solution: $g = argmin_{i=1..Np}(F_i)$

Training step:

Repeat until Condition

For all particle i **Do**

Update the velocity v_i and the position χ_i by (1) and (2);

Check if the values of the position vector are within the limitations of the search space.

Evaluate the fitness function F_i.

1. Decode the particle position to get C, epsilon, and gamma.
2. Build SVR model using the obtained C, epsilon, and gamma.
3. Fit the SVR model using the training dataset.
4. Predict the test dataset using the trained SVR model.
5. Calculate the mean squared errors to evaluate the performance of the SVR model.

Update the best personal position and its fitness value:

If Fi > Fpi **Then**

$Fp_i = F_i$

$p_i = \chi_i$

End If

End For

Update the global best position index: $g = argmin_{i=1..Np}(Fp_i)$

End repeat

End

We compared our proposed method (SVR-PSO) with SVR-GA, which is detailed in **SVR_GA Algorithm:**

In the crossover step, A two-point crossover function is chosen, where two crossover points are randomly selected along the length of the individuals. These points determine the segments of the individuals that will be swapped between the parents to create two offspring, which have the same length as the parents.

In the mutation step, we choose the bit flips function. The values of attributes in an input sequence are flipped to create a mutant individual. For Boolean individuals, this means changing a 0 to a 1 or vice versa. Each attribute (or gene) has an independent probability of being flipped.

In the selection step, the tournament function is employed, which is a method to select n individuals for reproduction. The tournament function picks the best individual from randomly chosen individuals, repeating the process "k" times. This process helps maintain diversity in the selected individuals. Fitness value is used as the selection criterion.

In the replace step, the next generation population is selected from both the offspring and the population.

SVR_GA Algorithm

Input: DB, GA parameters

Output: best SVR hyper parameters (C, epsilon, gamma)

Begin

Initializing step:

Random uniform Initialization of population within space search limitations.

Training step:

Repeat until condition:

Compute the value of the fitness function Fi of each individual x in the population.

1. Decode the particle position to get C, epsilon, and gamma.
2. Build SVR model using the obtained C, epsilon, and gamma.
3. Fit the SVR model using the training dataset.
4. Predict the test dataset using the trained SVR model.
5. Calculate the mean squared errors to evaluate the performance of the SVR model.

Selection

Crossover

Mutation

Replace

End repeat

End

4 RESULTS AND ANALYSIS

The proposed method and the comparative methods were implemented using the Python programming language and tested on the **Colab platform**.

4.1 Dataset description

The dataset employed in the test section contains 163 samples with five input parameters (biosorbent dose, temperature, pH, contact time, and dye concentration) and one output parameter (dye removal percentage R%). It originates from a study on Methylene Green (MG) adsorption conducted on Helianthus annuus seed shells (HA-Ss) powder, as detailed in the work of [32]. The bio-adsorbent, derived from Helianthus annuus seed shells, underwent thorough preparation to ensure its effectiveness in various adsorption applications. Initially, the shells were meticulously collected and subjected to multiple washes with distilled water. Subsequently, they were air-dried over a 10-day period. Once dried, the samples were finely powdered and sifted to achieve a uniform consistency. To further enhance its purity and stability, the powdered material underwent drying in an oven set at 100°C, followed by storage in desiccators until use. Prior to experimental application, the dried shells underwent additional processing, including fine grinding and sieving, to isolate particles within the diameter range of 250 to 315 μm, thus optimizing their suitability for adsorption testing. Batch tests were conducted at room temperature (25±0.2°C). The batch adsorption experiments involved mixing 20 mL of MG solution with HA-Ss powder in glass container tubes of 25 ml. To evaluate the adsorption capacity of MG from distilled water using the biosorbent, various

parameters were assessed, including biosorbent dose (ranging from 0.2 to 5 g/L), temperature (ranging from 25 to 45 °C), pH (ranging from 1 to 10), contact time (ranging from 5 to 120 minutes), and MG concentration (ranging between 10 to 50 mg/L). At each stage of experimentation, the adsorbents were separated from the aqueous medium using a centrifuge, and the MG concentration was determined using a UV-VIS spectrophotometer.

The efficiency (%) and capacity (q_e, mg/g) of MG removal using the HA-Ss adsorbent were evaluated using equations (4) and (5), respectively:

$$R\% = \frac{MG_i - M_f}{MG_i} \times 100 \quad (8)$$

$$q_e = \frac{MG_i - MG_f}{m} \times v \quad (9)$$

Where: MG_i and MG_f represent the initial and the final concentrations (mg/L) of MG dye, V is the volume (L) of the tested solution, and m is the mass (g) of the adsorbent.

4.2 Metrics

The performances of the proposed method SVR-PSO and the comparative methods SVR-GA and SVR-SG models were statistically measured using R^2 values and MSE as follows:

$$R^2 = 1 - \sum_{i=1}^{n} \left(\frac{(y_{\text{pred},i} - y_{\text{exp},i})^2}{(y_{\text{pred},i} - y_m)^2} \right) \quad (10)$$

$$\text{MSE} = \frac{\sum_{i=1}^{n} (y_{\text{pred},i} - y_{\text{exp},i})^2}{n} \quad (11)$$

where n is the number of experimental data, $y_{\text{pred},i}$ and $y_{\text{exp},i}$ are the predicted and experimental responses, respectively, and y_m is the average of experimental values. R^2 measures the percentage of total variation in the response variable that is explained by the least-squares regression. R^2 should be close to 1.0.

4.3 PARAMETERS SETTING

4.3.1 PSO parameters setting

We used the default values of the acceleration coefficients (c1 and c2) and the inertia weight parameter W, which are 2, 2, and 0.9, respectively, in the basic PSO algorithm.

The number of particles directly impacts the computational cost of running the PSO algorithm. Larger swarms require more memory and processing power. To determine the optimal number of particles, we tested the SVR-PSO algorithm with 10 and 20 particles. We observed that increasing the number of particles from 10 to 20 resulted in a longer running time without a corresponding improvement in algorithm performance.

The number of iterations in a Particle Swarm Optimization (PSO) algorithm plays a crucial role in determining the convergence behaviour and overall performance. Increasing the number of iterations allows the PSO algorithm more time to explore the search space and converge towards optimal or near-optimal solutions. However, a higher number of iterations also increases the computational cost of running the PSO algorithm. Each iteration involves evaluating the objective function for each particle and updating their positions and velocities. Therefore, there's a trade-off between the number of iterations and computational resources. A range of 20 to 50 iterations are tested. We noted that increasing the number of iterations from

20 to 50 resulted in a longer running time without a corresponding improvement in algorithm performance.

4.3.2 GA parameters setting

We set the following parameters for the genetic algorithm: the number of individuals to 20, the number of individuals to select for the next generation to 10, the number of children to produce at each generation to 20, the number of generations to produce to 10, the probability of mating two individuals to 0.7, the probability of mutating an individual to 0.3, the independent probability for each attribute to be flipped to 0.05, the number of individuals to select to 3, and the number of individuals participating in each tournament (tournsize=3).

4.3.3 Research space limitations

The Table 1 represents the lower and upper value limitations of each SVR hyperparameter.

Table 1 SVR Hyperparameters lower and upper values

Hyper parameter	Lower value	Upper value
C	200	900
Epsilon	1e-2	4e-2
Gamma	1e-2	2

4.4 Results and Discussion

We tested SVR-PSO, SVR-GA, and SVR-GS multiple times and summarized the best result of each in the Table 2.

Table 2 Results: SVR hyperparameters, Times, Metrics.

	SVR_GA	SVR_PSO	SVR_GS
C	559.6028	603.0539	734.0668
Epsilon	0.0264	0.0291	0.0261
Gamma	1.8902	1.4809	0.0323
Time (s)	3.4572	**1.4501**	26.0983
MSE	0.00032124	**0.00030082**	0.00143782
R^2	0.95319	**0.9562**	0.7905

We observed that: firstly, SVR-PSO and SVR-GA provided similar values for hyperparameters, unlike SVR-GS, particularly in the gamma hyperparameter. Secondly, SVR-PSO performed the best in terms of both time and performance, as indicated by the MSE value and R^2. The obtained results confirm that using PSO to tune the SVR hyperparameters yields better performance and speed.

5 Conclusion

The paper proposes a hybrid method, SVR-PSO, for modelling the dye removal process. Initially, Particle Swarm Optimization (PSO) is utilized to fine-tune the hyperparameters for Support Vector Regression (SVR). Subsequently, SVR is constructed using optimized hyperparameters and fitted to the data. The effectiveness of this method is assessed in modelling the removal of Malachite Green dye using Helianthus Annuus seed shells as an eco-friendly adsorbent. A comparison is made with SVR tuned using Genetic Algorithm (SVR-GA) and Grid Research method (SVR-GS). The results demonstrate that employing PSO for SVR hyperparameter tuning yields superior performance and faster optimization compared to GA and GR methods. SVR-PSO and SVR-GA produce similar hyperparameter values, underscoring their efficacy, while SVR-GS exhibits disparities, particularly in the gamma hyperparameter. Furthermore, SVR-PSO surpasses SVR-GA and SVR-GR in terms of both performance metrics and computational efficiency. Therefore, utilizing PSO for SVR hyperparameter tuning is recommended to enhance model accuracy and efficiency in similar applications.

REFERENCES

[1] P. P. Micklin, “Man and the water cycle: challenges for the 21st century,” GeoJournal, vol. 39, no. 3, pp. 285–298, Jul. 1996, doi: 10.1007/BF00188375.

[2] L. Petersen, M. Heynen, and F. Pellicciotti, “Freshwater Resources: Past, Present, Future,” in International Encyclopedia of Geography, John Wiley & Sons, Ltd, 2017, pp. 1–11. doi: 10.1002/9781118786352.wbieg0712.

[3] K. Singh and S. Arora, “Removal of Synthetic Textile Dyes From Wastewaters: A Critical Review on Present Treatment Technologies,” Crit. Rev. Environ. Sci. Technol., vol. 41, no. 9, pp. 807–878, Apr. 2011, doi: 10.1080/10643380903218376.

[4] A. Nasar and F. Mashkoor, “Application of polyaniline-based adsorbents for dye removal from water and wastewater—a review,” Environ. Sci. Pollut. Res., vol. 26, no. 6, pp. 5333–5356, Feb. 2019, doi: 10.1007/s11356-018-3990-y.

[5] V. Katheresan, J. Kansedo, and S. Y. Lau, “Efficiency of various recent wastewater dye removal methods: A review,” J. Environ. Chem. Eng., vol. 6, no. 4, pp. 4676–4697, Aug. 2018, doi: 10.1016/j.jece.2018.06.060.

[6] S. Dutta, B. Gupta, S. K. Srivastava, and A. K. Gupta, “Recent advances on the removal of dyes from wastewater using various adsorbents: a critical review,” Mater. Adv., vol. 2, no. 14, pp. 4497–4531, Jul. 2021, doi: 10.1039/D1MA00354B.

[7] I. Ali and V. K. Gupta, “Advances in water treatment by adsorption technology,” Nat. Protoc., vol. 1, no. 6, pp. 2661–2667, Dec. 2006, doi: 10.1038/nprot.2006.370.

[8] Md. Ahmaruzzaman, “Adsorption of phenolic compounds on low-cost adsorbents: A review,” Adv. Colloid Interface Sci., vol. 143, no. 1, pp. 48–67, Nov. 2008, doi: 10.1016/j.cis.2008.07.002.

[9] M. R. Adam et al., "Advances in adsorptive membrane technology for water treatment and resource recovery applications: A critical review," J. Environ. Chem. Eng., vol. 10, no. 3, p. 107633, Jun. 2022, doi: 10.1016/j.jece.2022.107633.

[10] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects," J. Ind. Inf. Integr., vol. 23, p. 100224, Sep. 2021, doi: 10.1016/j.jii.2021.100224.

[11] R. Priyadarshini, R. M. Mehra, A. Sehgal, and P. J. Singh, Artificial Intelligence: Applications and Innovations. CRC Press, 2022.

[12] A. M. Ghaedi and A. Vafaei, "Applications of artificial neural networks for adsorption removal of dyes from aqueous solution: A review," Adv. Colloid Interface Sci., vol. 245, pp. 20–39, Jul. 2017, doi: 10.1016/j.cis.2017.04.015.

[13] G. Alam, I. Ihsanullah, Mu. Naushad, and M. Sillanpää, "Applications of artificial intelligence in water treatment for optimization and automation of adsorption processes: Recent advances and prospects," Chem. Eng. J., vol. 427, p. 130011, Jan. 2022, doi: 10.1016/j.cej.2021.130011.

[14] V. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation and signal processing," in Proceedings of the 9th International Conference on Neural Information Processing Systems, in NIPS'96. Cambridge, MA, USA: MIT Press, Dec. 1996, pp. 281–287.

[15] V. N. Vapnik, The Nature of Statistical Learning Theory. New York, NY: Springer, 2000. doi: 10.1007/978-1-4757-3264-1.

[16] M. Awad and R. Khanna, "Support Vector Regression," in Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, M. Awad and R. Khanna, Eds., Berkeley, CA: Apress, 2015, pp. 67–80. doi: 10.1007/978-1-4302-5990-9_4.

[17] W. Zhao, T. Tao, and E. Zio, "System reliability prediction by support vector regression with analytic selection and genetic algorithm parameters selection," Appl. Soft Comput., vol. 30, pp. 792–802, May 2015, doi: 10.1016/j.asoc.2015.02.026.

[18] X. Ma, Y. Zhang, and Y. Wang, "Performance evaluation of kernel functions based on grid search for support vector regression," in 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Jul. 2015, pp. 283–288. doi: 10.1109/ICCIS.2015.7274635.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, Nov. 1995, pp. 1942–1948 vol.4. doi: 10.1109/ICNN.1995.488968.

[20] Z. Hu, M. Wang, and X. Huang, "Parameter Selection of Support Vector Regression Based on Particle Swarm Optimization," 2010 IEEE Int. Conf. Granul. Comput., pp. 834–838, Aug. 2010, doi: 10.1109/GrC.2010.121.

[21] M. Mitchell, An Introduction to Genetic Algorithms. 1998. [Online]. Available: https://mitpress.mit.edu/9780262631853/an-introduction-to-genetic-algorithms/

[22] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Cornélio, "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," Inf. Softw. Technol., vol. 52, no. 11, pp. 1155–1166, Nov. 2010, doi: 10.1016/j.infsof.2010.05.009.

[23] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," J. Glob. Optim., vol. 11, no. 4, pp. 341–359, Dec. 1997, doi: 10.1023/A:1008202821328.

[24] R. M. Hlihor, M. Diaconu, F. Leon, S. Curteanu, T. Tavares, and M. Gavrilescu, "Experimental analysis and mathematical prediction of Cd(II) removal by biosorption using support vector machines and genetic algorithms," New Biotechnol., vol. 32, no. 3, pp. 358–368, May 2015, doi: 10.1016/j.nbt.2014.08.003.

[25] S. Mandal, S. S. Mahapatra, S. Adhikari, and R. K. Patel, "Modeling of Arsenic (III) Removal by Evolutionary Genetic Programming and Least Square Support Vector Machine Models," Environ. Process., vol. 2, no. 1, pp. 145–172, Mar. 2015, doi: 10.1007/s40710-014-0050-6.

[26] M. Ghaedi, M. reza Rahimi, A. M. Ghaedi, I. Tyagi, S. Agarwal, and V. K. Gupta, "Application of least squares support vector regression and linear multiple regression for modeling removal of methyl orange onto tin oxide nanoparticles loaded on activated carbon and activated carbon prepared from Pistacia atlantica wood," J. Colloid Interface Sci., vol. 461, pp. 425–434, Jan. 2016, doi: 10.1016/j.jcis.2015.09.024.

[27] A. Asfaram, M. Ghaedi, M. H. A. Azqhandi, A. Goudarzi, and M. Dastkhoon, "Statistical experimental design, least squares-support vector machine (LS-SVM) and artificial neural network (ANN) methods for modeling the facilitated adsorption of methylene blue dye," RSC Adv., vol. 6, no. 46, pp. 40502–40516, Apr. 2016, doi: 10.1039/C6RA01874B.

[28] N. Parveen, S. Zaidi, and M. Danish, "Support vector regression (SVR)-based adsorption model for Ni (II) ions removal," Groundw. Sustain. Dev., vol. 9, p. 100232, Oct. 2019, doi: 10.1016/j.gsd.2019.100232.

[29] M. Solgi, T. Najib, S. Ahmadnejad, and B. Nasernejad, "Synthesis and characterization of novel activated carbon from Medlar seed for chromium removal: Experimental analysis and modeling with artificial neural network and support vector regression," Resour.-Effic. Technol., vol. 3, no. 3, pp. 236–248, Sep. 2017, doi: 10.1016/j.reffit.2017.08.003.

[30] M. Gholami Shirkoohi, R. D. Tyagi, P. A. Vanrolleghem, and P. Drogui, "A comparison of artificial intelligence models for predicting phosphate removal efficiency from wastewater using the electrocoagulation process," Digit. Chem. Eng., vol. 4, p. 100043, Sep. 2022, doi: 10.1016/j.dche.2022.100043.

[31] M. Ghaedi, K. Dashtian, A. M. Ghaedi, and N. Dehghanian, "A hybrid model of support vector regression with genetic algorithm for forecasting adsorption of malachite green onto multi-walled carbon nanotubes: central composite design optimization," Phys. Chem. Chem. Phys., vol. 18, no. 19, pp. 13310–13321, May 2016, doi: 10.1039/C6CP01531J.

[32] H. Fiala et al., "Removal of Malachite Green Dye From Aqueous Media using Helianthus Annuus Seeds Shells as Eco-Friendly Adsorbent: Optimization, Equilibrium, and Kinetic Studies," Ann. Romanian Soc. Cell Biol., vol. 25, no. 6, Art. no. 6, Jul. 2021.

[33] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. 2010. Accessed: Nov. 15, 2023. [Online]. Available: https://www.semanticscholar.org/paper/An-Introduction-to-Support-Vector-Machines-and-Cristianini-Shawe-Taylor/5c04f8002e24a8c09bfbfedca3c6c346fe1e5d53

[34] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, 1975.