Ensemble Stacking Case-Based Reasoning for Regression

Daniel Soto-Forero[1 [0000−0003−0753−4673]], Marie-Laure Betbeder[1 [0000−0002−8103−4098]], and Julien Henriet[1 [0000−0002−7671−4574]]

DISC, Université de Franche-Comté, CNRS, institut FEMTO-ST, 16 Route de Gray, Besançon, F-25000, France.
{daniel.soto_forero, marie-laure.betbeder, julien.henriet}@univ-fcomte.fr

Abstract. This paper presents a case-based reasoning algorithm with a two-stage iterative double stacking to find approximate solutions to one and multidimensional regression problems. This approach does not require training, so it can work with dynamic data at run time. The solutions are generated using stochastic algorithms in order to allow exploration of the solution space. The evaluation is performed by transforming the regression problem into an optimization problem with an associated objective function. The algorithm has been tested in comparison with nine classical regression algorithms on ten different regression databases extracted from the UCI site. The results show that the proposed algorithm generates solutions in most cases quite close to the real solutions. According to the RMSE, the proposed algorithm globally among the four best algorithms, according to MAE, to the fourth best algoritms of the ten evaluated, suggesting that the results are reasonably good.

Keywords: Case-Based Reasoning · Stacking · Regression · Ensemble Methods · Machine Learning

1 Introduction

One of the main issues addressed by case-based reasoning (CBR) is its ability to adapt solutions from source cases to the target ones [16]. Choosing the best adaptation strategy can be based on the prediction of its ability to solve and align with the problem. Thus, its predictive ability can be used to solve a regression problem. In addition, unanticipated events occur during the application of a solution. These events lead the human operator to adapt the proposed solution in real-time quickly.

Ensemble methods use several multiple models performed independently, and their results are combined to obtain a final global prediction. The main idea is to improve the results and generalization ability of individual models [1]. Some ensemble methods use different models with different sets of data; others use the models with the same data but different parameters. The combination of the results of the multiple models can use different strategies as simple rules or more

complex approaches [2]. The ensemble methods are useful in classification and regression problems.

Machine learning methods can be applied to different types of regression problems to predict values by building, evaluating, and training complex linear and non-linear models. However, if it is possible to improve accuracy by integrating them, the most common integration strategies used on ensemble learning are stacking, boosting and bagging [12]. Stacking is a kind of ensemble deep learning meta-learning model, whose purpose is to use various machine learning techniques to overcome the limitations of individual models. The integration generates a final result with improved accuracy [4]. In stacking methods, the base algorithms are called level-0. Generally, they are heterogeneous machine learning models or algorithms, and each works with the same database. The meta algorithm unifying the results from level-0 algorithms using another machine-learning techniques or a set of rules is called level-1 [11].

In this paper, we present a new method for regression based on the ensemble approach combined with the CBR one. Our proposition consists of the enrichment of the knowledge containers defined by M. Richter [15] using the ensemble approach. This paper is organized as follows: Section II presents the related works about case-based reasoning, ensemble techniques, and regression. The proposed model is explained in Section III. The Section IV shows the experimental description and the results. The results are discussed in section Section V. Finally, the conclusions and future work are discussed in Section VI.

2 Related Works

Different strategies for retrieval and adaptation in CBR-systems are proposed in the literature. For example, S. Petrovic *et al.* [13] proposed to use neural networks, Jung *et al.* [7] k-means clustering mixed with RBFN, R. Reyes *et al.* [16] propose a mix with CSP (Constraint satisfaction problems), Y. Lepage *et al.*[9] proposed an alternative retrieve phase based on the LCS (longest common subsequence) metric. Uysal *et al.* [18] implement a CBR with a bootstrap aggregation method (bagging) to improve the CBR accuracy and reduces the variance. D. Leake *et al* [8] conducted a study analyzing the potential benefits of combining deep learning with CBR to enhance overall performance. All these strategies inspired the work presented in this paper and lead us to the possibility to solve regression problems using CBR. In addition, we studied Ensemble methods and stacking and integrate them to our CBR-based proposition.

2.1 Ensemble Methods

Ju *et al.* [6] study a robust non-parametric regression based on gradient boosting using a linear combination of base learners in two stages. The authors identify

the limitations of a classical gradient algorithm: the loss function generally must be convex, but if that is not the case, then the algorithm may not work well. The algorithm behavior is different and has very low efficiency when the data does not contain outliers and, sometimes, the overfitting in the training step. To mitigate these problems, an ensemble model is proposed that combines gradient regression with tree regression in two stages to correct the initial estimator and define an early stopping time with a function that stops the training if the validation error is static after several iterations. The algorithm has been tested in comparison with other boosting regression methods over six database configurations generated with three formula bases, using the RMSE as a metric. The results confirm the good performance and accuracy of the ensemble methods, but only if the good combination of algorithms and aggregation of results is well done.

An ensemble Case-Based Reasoning model is proposed by Yu *et al.* [19] applied to financial prediction and fill missing data. In this case, to retrieve the nearest neighbors, the model uses three different distance metrics and an integration voting stage. The model has been tested with a database with eleven dimensions of financial information from 249 enterprises. The comparison is made with two objectives. First, the filling in missing data with other algorithms like KNN or RandomForest, and second, the prediction comparison with single algorithms using a specific distance metric. Indeed, the results show better performance filling in the missing data and the highest results in prediction.

2.2 Stacking

The work of Mang *et al.* [12] uses stacking method integrating support vector regression (SVR), kernel ridge regression (KRR), and elastic net (ENET) algorithms for prediction in three large genomic datasets. The proposed stacking method is compared with the single algorithms SVR, KRR, ENET, and BayesB using a precision metric over three regression databases. The meta-algorithm for level-1 is an ordinary least squares linear regression (OLS). The algorithms are performed with a 20-fold cross-validation.

Some machine learning algorithms are used for the automatic extraction of building information from images, but Cao *et al.* [3] propose an ensembling method with three neural network models (U-net, SegNet, and FCN-8), whose role is to extract the building features. Each model is optimized with a conditional random fields algorithm, and then all model results are combined with a method based on a sparse autoencoder. The algorithm was evaluated with a database created with 650 satellite images of various Chinese cities with an approximate resolution of 5000x3500 pixels. The algorithm has been compared with the single network models, and the results show that the ensemble method obtains better accuracy, recall, and F1 score than the individual algorithms. Some identified limitations are the computation time, memory, and resources,

but it is possible to improve the stacking model to correct these limitations.

Evolutionary iterative techniques can also be used in stacking models as shown in Bakurov *et al.* [2], where the stacking is used to study the impact of changes in genetic programming parameters and improve the genetic programming systems. The focus study is the initialization and selection phases as well as the used operators and stopping criteria. The base learners in this case are multi-linear regression, multi-layer perceptron, random forest regression and support vector regression. Genetic programming has the meta-learning algorithm role in the stacking model. The test was realized with 11 regression databases, seven synthetic and four real-world problems. The experiment was run 60 times with different data partitions. The results yield the optimal hyper-parameters to the genetic programming algorithm according to the best result obtained with MSE and average rank metrics.

3 Proposed Model

The proposed algorithm ESCBR (ensemble stacking case-based reasoning) is based on the generic CBR paradigm associated with several neighbor search and solution generation algorithms that have been integrated according to a stacking model variation in two iterative stages. The integration with the stacking model gives the algorithm the ability to adapt itself to different types of problems, and avoid biases and overtraining. The results of the execution of the stacking levels store knowledge containers in the CBR memory[15] and then help the learning of the algorithm iteratively. In addition, this knowledge container facilitates the generation of solutions to various problems in different databases without the need for a preliminary training phase. Iterative design in two cycles improves the capacity of the CBR system to work and adapt to dynamic problems at run time.

Figure 1 presents the links between the knowledge containers of our CBR system and the different phases of this process. Figure 2 presents the complete workflow of the proposed algorithm. The variables and parameters for the proposed algorithm are shown in Table 1. As shown in these two figures and this table, the retrieve stage uses the search algorithms and the case database (containers C1 and C3) in order to find the nearest neighbors of a given new problem. The reuse stage uses the solution generation algorithms (container C2). The revise stage evaluates the generated solutions and allows the generation of new solutions iteratively according to the parameters stored in the C4 container. The systems invoke the reconfigurate stage in order to change the combination of algorithms. With a selected solution, the renovate stage is called to update the parameters and the container data. Finally, in the retain stage, the case base is updated with the new case and the generated solution.

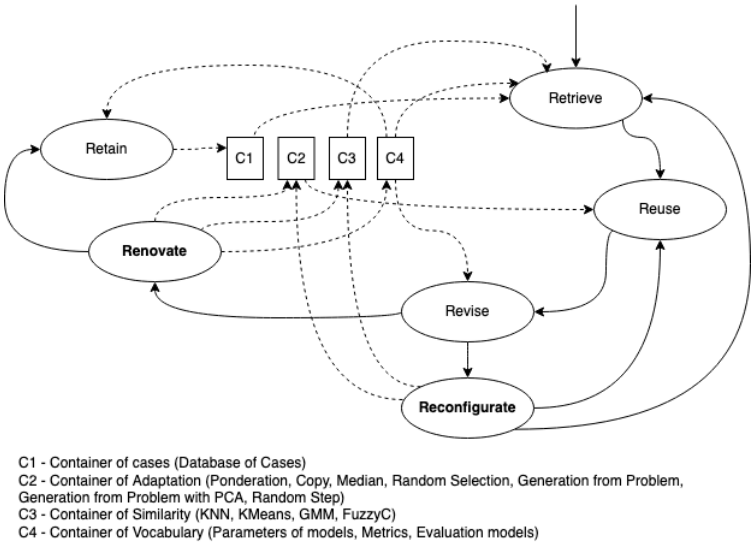

Fig. 1. Two cycles of proposed CBR

Table 1. Variables and parameters of proposed model (Type: p - parameter, v - variable, f - function)

ID	Type	Description	Domain
it	p	Number of iterations	$\mathbb{N}, it > 0$
np	p	Number of process	$\mathbb{N}, np > 2$
nl	p	Maximum number of local neighbors	$\mathbb{N}, nl > 0$
ng	p	Number of global neighbors	$\mathbb{N}, ng > 2$
n	v	Dimension of problem space	$\mathbb{N}, n > 0$
m	v	Dimension of solution space	$\mathbb{N}, m > 0$
z	v	Database size	$\mathbb{N}, z > 0$
p	v	Problem description	$\mathbb{R}^n$
s	v	Solution description	$\mathbb{R}^m$
r_a	v	Number of Retrieve Models	$\mathbb{N}, r_a > 2$
r_b	v	Number of Reuse Models	$\mathbb{N}, r_b > 2$
at	v	Actions	$[0, 2] \in \mathbb{N}$
nl_i	v	Number of local neighbors for i model	$\mathbb{N}, nl_i \leq nl$
g	v	Global best solution description	$\mathbb{R}^m$
v	v	Global best solution evaluation	$\mathbb{R}$
$d(x_1, x_2)$	f	Distance function between x_1 and x_2	$\mathbb{R}$
$rn(x, y)$	f	Random value with Normal distribution x mean, y standard deviation	$\mathbb{R}_+$
$MP(x_1^z, x_2, a)$	f	Retrieve model function between x_1 and x_2	$\mathbb{R}^{a \times z}$
$MS(x_1^m)$	f	Reuse model function with x_1	$\mathbb{R}^m$
$f_s(p^n, s^m)$	f	Solutions evaluation	$\mathbb{R}$

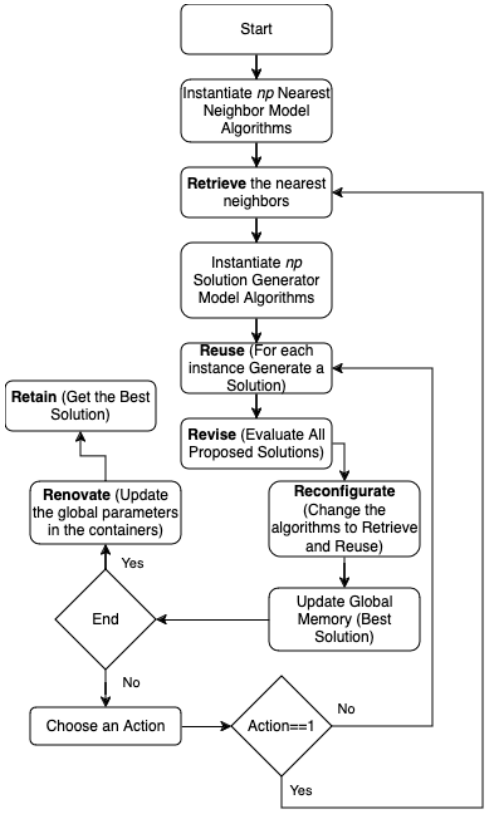

Fig. 2. Flow of stacking CBR

3.1 Retrieve

The first step of the algorithm consists of finding the most similar cases to a new eventual case. Figure 3 shows the different processes used in the stacking model. At level-0, each process selects and executes a different neighbor search algorithm chosen from r_a models into the C3 container, with a number of neighbors nl_i randomly chosen in the interval $[0, nl]$. Then, at level-1, the results are unified by building a global set of similar cases. Five algorithms have been implemented for the retrieve stage: KNN (K-nearest neighbors), K-means, GMM (Gaussian mixture model), fuzzyC-Means and weighted KNN.

KNN is a machine learning algorithm based on learning approach that sort the elements of a dataset considering nearby instances in the feature space [17], this algorithm can be parameterized to consider different weights for each dimension (weighted KNN). K-Means is an algorithm which calculates the sum of distances from each data point to a certain cluster center as optimization objective, and updates the cluster center by using the average value of samples

in the cluster. Its goal is to group similar points [10]. GMM is a model that uses a parametric probability density function in order to define a weighted sum of K-Gaussian component densities. The model can be used for regression, classification or clustering problems [14]. FuzzyC-Means is also a clustering algorithm that introduces the fuzzy set theory to quantify the cluster membership uncertainty [5].

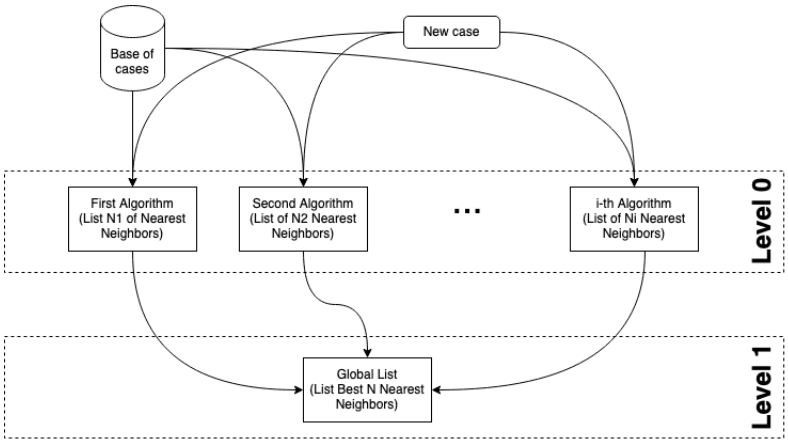

Fig. 3. Stacking for nearest neighbors search

Formally, the first proposed stacking model works with a database of z cases, where a case is composed of the problem description and the solution description $(p^n, s^m)^z$ and a target case (without solution p_w^n). The goal of all level-0 algorithms is to generate a local list of cases similar to the new case using the problem description information. Thus, a set $X_j = \{x_1, x_2, ..., x_z \mid x_i = MP_i((p^n)^z, p_w^n, nl_i)\}$ is generated considering each j model performed. At level-1, a global set $X_g = \biguplus_{n=1}^{ng} min_{distance}\ ((\cup_{j=1}^{np} X_j))$ using all the j local sets is created, where $\biguplus$ represents an exclusive union (union of elements without repetition). Then, the result of the first stacking model is the X_g set with the ng global nearest neighbors.

3.2 Reuse

Once the global list of similar cases has been constructed, the information corresponding to the solutions of each of those cases is extracted and used to generate a new solution that adapts to the new case using similar cases and similar solutions as shown in Figure 4. All the generation algorithms must respect the generation rules defined at the beginning of the stacking process. These rules define the restrictions of certain attributes that the generated solutions must have.

If no rules are defined, then the algorithms can manipulate the attributes of the solutions freely. The generation is performed with a second stacking model with different processes, as shown in Figure 5. At level-0, each process selects and executes a different generation algorithm from the r_b models into the C2 container. At level-1, all the different solutions generated are stored in a global memory. Ten algorithms have been implemented for the reuse stage in level-0: weighting with probability, weighting without probability, median values, Copy/Change, voting, interpolation, PCA (principal component analysis), and random step.

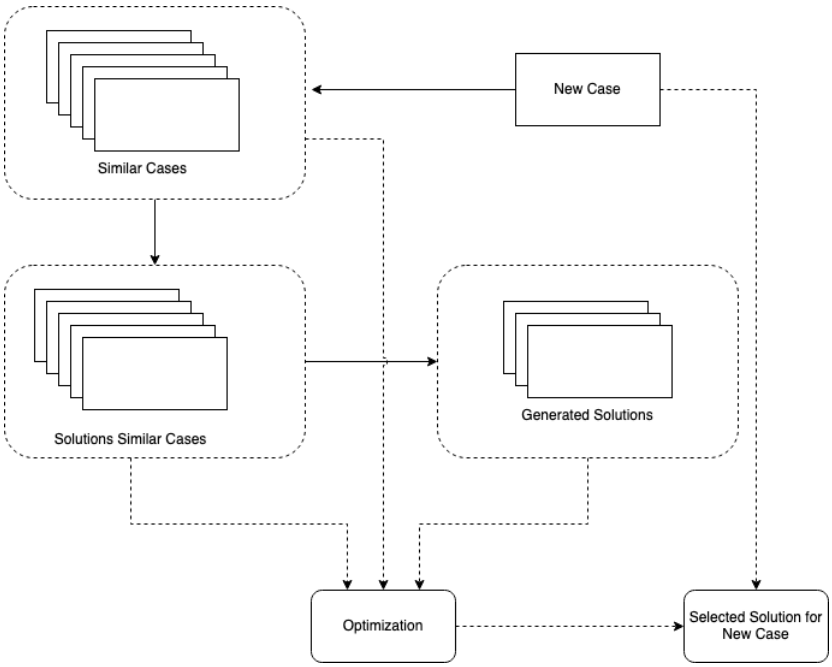

Fig. 4. Automatic generation and verification of solutions

The process that implements the weighting random selection with the probability algorithm builds a solution copying randomly the information of solutions with bigger probability to the associated nearest problem case. The process that executes weighting random selection without the probability algorithm randomly copies the information of solutions according to the uniform distribution. The process that computes median values uses the median value of all the solutions for each dimension. The copy/change-based process copies the information of a random solution and changes a portion with the information of another randomly selected solution. The voting-based process copies the information that is most frequent between the solutions. The interpolation-based process uses random information from a calculated interpolation function. The process based on

PCA transforms the problem description to the space of solutions description to establish a relation between the problem and its solution, through a distance metric, with which a possible solution to the new problem can be inferred, considering the solutions that lie within the radius of the distance obtained. The process that executes the random step algorithm chooses a solution and changes the values in one dimension randomly with a small step.

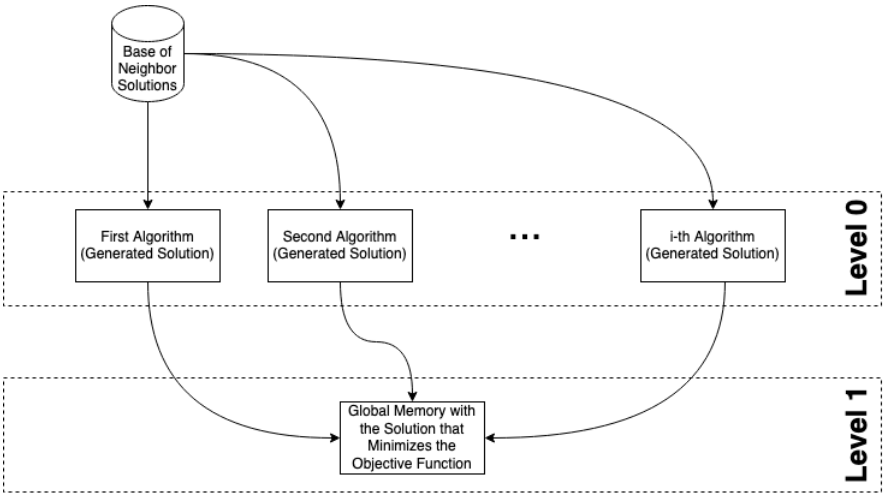

Fig. 5. Stacking for solution generation

This second stacking model works with the solution description s as a parameter with the set $(s^m)^{ng}$. Each performed reuse model can generate a candidate solution $s_{i,c} = MS_i((s^m)^{ng})$. The level-1 is the build of a unification set of all candidate solutions $Y_g = \cup_{i=1}^{np} s_{i,c}$. This set is evaluated with a function to determine the quality of solution.

Finally, the candidate solutions are compared and evaluated. This problem is transformed into an optimization problem, where the objective function is 1.

$$min\ \ (f_s(p_w^n, s_w^m)) = min\left(\sum_{i=1}^{ng} \frac{d(s_w^m, s_i^t)}{d(p_w^n, p_i^n)^2}\right) \tag{1}$$

$$s_i^t = s_i^m + rn(0, d(p_w^n, p_i^n)) \tag{2}$$

The optimization cycle can execute the retrieve and reuse phases according to the selected random action from $[0, at]$ in each it iteration, saving inside a global memory in each iteration the solution that obtains the minimum value in

the evaluation of the objective function.

3.3 Reconfigurate

The internal cycle (retrieve, reuse and revise) has the possibility to dynamically change the combination of algorithms that can be executed in each stage for each process. During this reconfigurate step, information stored in the containers C2 and C3 are used in order to configure and instantiate the new selected algorithms. Each process can change its algorithms randomly during runtime, in any iteration, asynchronously.

3.4 Renovate

After having selected a solution from all the candidates proposed by the process, the renovate phase is executed, which updates the information (used algorithms and their respective parameters) of C2, C3 and C4 containers. This allows the system to learn according to the predictions made, and to propose better results during the next executions. The information inside the containers can be changed in order to fit to a specific problem as best as possible.

3.5 Revise and retain

As usual, in CBR systems, the revise phase is left to the system user. The retain stage simply takes the best-proposed solution and determines whether it is a new or existing solution and, if it is new, stores it in the database.

4 Results

In order to compare the performance prediction and behavior of the proposed algorithm, ten regression databases with different characteristics has been selected. The databases and their characteristics are shown in Table 2. The values used as parameters to our algorithm are: $it = 100$, $np = 50$, $nl = 10$ and $ng = 10$, the parameters for stacking algorithms are chosen randomly. They are adjusted to get a global convergence during the renovate phase.

The comparison of the proposed algorithm is made with respect to nine well-known regression algorithms widely used in various research and applied problems. The list of algorithms is shown in Table 3. The parameters for each algorithm are shown in Table 4. All the algorithms have been executed 10 times, and their data have been partitioned in k-folds with $k = 100$. The results have

Table 2. Description of evaluated datasets. (* After encoding String data)

ID	DataSet	Features	Instances	Output Dimension	Input Domain	Output Domain
DS1	Yatch Hydrodynamics	6	308	1	$\mathbb{R}$	$\mathbb{R}$
DS2	Electrical Grid Stability	12	10000	1	$\mathbb{R}$	$\mathbb{R}$
DS3	Real State Valuation	6	414	1	$\mathbb{R}_+$	$\mathbb{R}_+$
DS4	Wine Quality (Red)	11	1598	1	$\mathbb{R}_+$	$\mathbb{N}$
DS5	Wine Quality (White)	11	4897	1	$\mathbb{R}_+$	$\mathbb{N}$
DS6	Concrete Compressive Strength	8	1030	1	$\mathbb{R}_+$	$\mathbb{R}_+$
DS7	Energy Efficiency	8	768	2	$\mathbb{R}_+$	$\mathbb{R}_+$
DS8	Gas Turbine CO, NOx Emission (2015)	9	7384	2	$\mathbb{R}_+$	$\mathbb{R}_+$
DS9	Student Performace Portuguese	30	649	3	$\mathbb{N}*$	$\mathbb{N}$
DS10	Student Performance Math	30	395	3	$\mathbb{N}*$	$\mathbb{N}$

Table 3. List of evaluated algorithms

ID	Algorithm	ID	Algorithm
A1	Linear Regression	A6	Polinomial Regression
A2	K-Nearest Neighbor	A7	Ridge Regression
A3	Decision Tree	A8	Lasso Regression
A4	Random Forest (Ensemble)	A9	Gradient Boosting (Ensemble)
A5	Multi Layer Perceptron	A10	Proposed Case Based Reasoning

been calculated with the best results of ten executions.

Table 5 shows the detailed results and average ranking for all the databases according to the RMSE (root mean square error) metric. The gradient boosting algorithm (A9) achieves the global best-ranking value, and the proposed algorithm (A10) is placed in fourth position. Table 6 shows the detailed results of the same algorithms and the same databases but compared with the MAE (median absolute error) metric. In that case, the best global average value is for random forest algorithm (A4), and the proposed algorithm (A10) is placed globally in fourth position with the best results in databases DS4 and DS5.

Figure 6 shows the global dispersion, median, and outliers for four representative databases, where it can be seen that the proposed algorithm generates more outliers than other algorithms, but the variance is low and the convergence is close to the real value, better than most of the compared algorithms. These four databases are representative because they have very different characteristics and consider variables to be predicted in different dimensions.

Table 4. Parameters for all compared algorithms

ID	Parameter	Value	ID	Parameter	Value
A1	Intercept	True	A6	Degree	4
	Positive	True		Bias	True
A2	Neighbors	5	A7	Fit Intercept	True
	Weights	Uniform		alpha	0.2
	Metric	Minkowsky		tol	1e-4
	Power Minkowsky	2			
A3	Error	Squared Error	A8	Fit Intercept	True
	Min samples split	2		alpha	[0.00001, 0.4]
				Max iter	1000
				tol	1e-4
A4	Estimators	10	A9	Error	Squarred Error
	Error	Squared Error		Learning Rate	0.1
	Min samples split	2		Estimators	100
	Bootstrap	True		Min Split	2
A5	Hidden Layers	100			
	Activation	Relu			
	Solver	Adam			
	alpha	0.0001			
	Learning Rate	0.001			
	Max Iter	200			
	beta1	0.9			
	beta2	0.999			
	epsilon	1e-8			

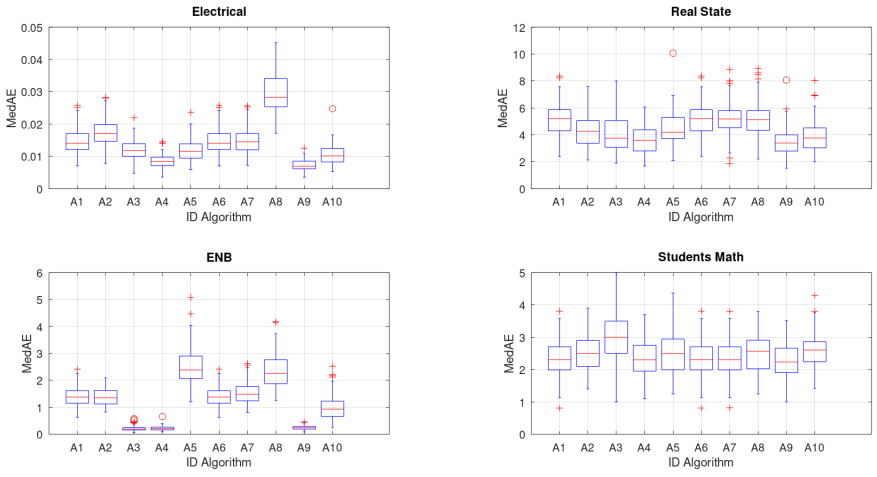

Fig. 6. Results of MAE (median absolute error) for ten algorithms and four representative databases

Table 5. Average ranking of best RMSE (Root Mean Squared Error) for all datasets with machine learning regression algorithms

Dataset	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
DS1	9.010	10.780	1.224	0.982	3.369	9.009	8.985	9.629	**0.668**	5.871
DS2	0.022	0.025	0.020	0.012	0.017	0.022	0.022	0.037	**0.011**	0.015
DS3	8.633	8.033	9.334	**7.203**	8.470	8.705	8.842	9.009	7.324	8.491
DS4	0.651	0.746	0.782	**0.571**	0.694	0.651	0.651	0.792	0.617	0.762
DS5	0.753	0.806	0.820	**0.599**	0.853	0.754	0.757	0.863	0.688	0.748
DS6	10.439	8.871	6.144	**4.738**	6.553	10.423	10.422	10.428	5.053	8.766
DS7	2.948	2.116	0.541	**0.465**	3.726	2.949	2.979	4.094	0.467	1.973
DS8	1.315	1.161	1.513	**1.109**	1.566	1.303	1.308	1.318	1.125	2.157
DS9	**2.304**	2.624	3.217	2.315	2.898	**2.304**	**2.304**	2.551	2.342	2.802
DS10	3.052	3.404	4.158	**3.014**	3.607	3.061	3.061	3.150	3.020	3.874
Avg. Rank	5.7	6.3	7.2	2.1	6.6	5.6	5.5	8.6	1.8	5.6

Table 6. Comparison of best MAE (median absolute error) for multiple datasets with machine learning regression algorithms

Dataset	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
DS1	6.776	2.385	0.231	0.207	3.632	6.778	6.307	5.186	**0.162**	1.193
DS2	0.015	0.017	0.012	0.008	0.012	0.015	0.015	0.030	**0.007**	0.011
DS3	5.092	4.320	4.1	3.632	4.435	5.092	5.20	5.132	**3.504**	3.90
DS4	0.413	0.495	0.18	0.325	0.451	0.413	0.412	0.544	0.387	**0.154**
DS5	0.509	0.548	0.285	0.374	0.550	0.509	0.509	0.633	0.456	**0.113**
DS6	6.989	5.709	3.134	**2.839**	4.306	6.989	6.989	6.986	3.084	5.439
DS7	1.393	1.372	**0.217**	0.218	2.523	1.393	1.529	2.346	0.243	1.008
DS8	0.549	0.297	0.365	**0.289**	0.742	0.549	0.549	0.540	0.309	0.861
DS9	**1.496**	1.788	2.080	1.612	2.005	**1.496**	**1.496**	1.714	1.538	1.721
DS10	2.344	2.534	2.910	2.331	2.543	2.344	2.344	2.481	**2.258**	2.602
Avg. Rank	6.45	6.4	4.35	2.3	7.35	6.55	6.6	7.9	2.4	4.7

5 Discussion

The proposed algorithm reveals a competitive performance in comparison with some of the most popular, most used, and recent algorithms for prediction in regression problems. Specifically, in this work, we have run the tests on ten databases with different characteristics, such as the number of instances, the number of features, the domain of the input variables, the dimensions of the output variable, and the subject area demonstrating the versatility of the proposed algorithm and the applicability to different configurations. Given the exploratory and stochastic nature of the proposed algorithm, it presents a great diversity of solutions generating several outliers; but despite, this in most cases, it is possible to reach an approximate solution that converges close to the real solution. It is the reason why in some cases the values of the mean are high but with the median remains low.

It can also be seen that the integration of the search algorithms produces better results than simple algorithms, as in the case of the proposed algorithm compared to KNN or compared to Linear Regression; although, for the proposed algorithm the impact of the first and second stacking on the final results obtained has not been determined exactly.

Globally, for the RMSE, the boosting algorithms perform better overall than the classical algorithms, even though the performance is variable. The proposed algorithm obtains acceptable values in all the databases. According to the average of the ranking positions, it is placed in the fourth place for RMSE, and the algorithm is placed in the fourth place for MAE.

An important aspect of the proposed algorithm is the objective function, which could be evaluated and modified dynamically depending on the characteristics of the evaluated problem, given that in the present study, the tests have been performed with the intuitive function that provides a greater probability of selection and evolution to the solution associated to the nearest neighbors; but, it is possible to complement the evaluation with other relevant terms and in that way improve the results.

In addition to the results, the proposed algorithm presents several advantages with respect to the algorithms with which it has been compared. Among these advantages, it does not require training, it can integrate algorithms and rules in each stacking, and by the design in two cycles, it can work with dynamic problems at runtime.

6 Conclusion

This paper proposes a generic regression technique using case-based reasoning and a stacking model, whose main characteristics are that it does not require training, and that due to the internal iterative cycle, it can adapt to dynamic problems in real time. The numerical results obtained in the tests performed show the potential of the algorithm with varied data and databases of different sizes, as well as the competitiveness with other standard and robust algorithms commonly used in regression problems.

As future work, it is envisaged to use the proposed model in an intelligent learning system to complement the real-time recommendation modules, since the algorithm does not require training and adapts dynamically to the data. Another important modification is to integrate the ESCBR with multi agent systems to improve the performance and precision, reduce the variance, and avoid the generation of outliers. Additionally, further studies could include various retrieve techniques to locate neighbors according to some rules within a given context.

References

1. Ali, M., Haque, M.u., Durad, M.H., Usman, A., Mohsin, S.M., Mujlid, H., Maple, C.: Effective network intrusion detection using stacking-based ensemble approach. International Journal of Information Security **22** (2023). https://doi.org/10.1007/s10207-023-00718-7, https://doi.org/10.1007/s10207-023-00718-7
2. Bakurov, I., Castelli, M., Gau, O., Fontanella, F., Vanneschi, L.: Genetic programming for stacked generalization. Swarm and Evolutionary Computation **65**, 100913 (2021). https://doi.org/https://doi.org/10.1016/j.swevo.2021.100913, https://www.sciencedirect.com/science/article/pii/S2210650221000742
3. Cao, D., Xing, H., Wong, M.S., Kwan, M.P., Xing, H., Meng, Y.: A stacking ensemble deep learning model for building extraction from remote sensing images. Remote Sensing **13**(19) (2021). https://doi.org/10.3390/rs13193898, https://www.mdpi.com/2072-4292/13/19/3898
4. Choi, J., Suh, D., Otto, M.O.: Boosted stacking ensemble machine learning method for wafer map pattern classification. Computers, Materials & Continua **74**(2), 2945–2966 (2023). https://doi.org/10.32604/cmc.2023.033417, http://www.techscience.com/cmc/v74n2/50296
5. Jiao, L., Yang, H., ga Liu, Z., Pan, Q.: Interpretable fuzzy clustering using unsupervised fuzzy decision trees. Information Sciences **611**, 540–563 (2022). https://doi.org/https://doi.org/10.1016/j.ins.2022.08.077, https://www.sciencedirect.com/science/article/pii/S0020025522009872
6. Ju, X., Salibián-Barrera, M.: Robust boosting for regression problems. Computational Statistics & Data Analysis **153**, 107065 (2021). https://doi.org/https://doi.org/10.1016/j.csda.2020.107065, https://www.sciencedirect.com/science/article/pii/S0167947320301560
7. Jung, S., Lim, T., Kim, D.: Integrating radial basis function networks with case-based reasoning for product design. Expert Systems with Applications **36**(3, Part 1), 5695–5701 (2009). https://doi.org/https://doi.org/10.1016/j.eswa.2008.06.099, https://www.sciencedirect.com/science/article/pii/S0957417408003667
8. Leake, D., Crandall, D.: On bringing case-based reasoning methodology to deep learning. In: Watson, I., Weber, R. (eds.) Case-Based Reasoning Research and Development. pp. 343–348. Springer International Publishing, Cham (2020)
9. Lepage, Y., Lieber, J., Mornard, I., Nauer, E., Romary, J., Sies, R.: The french correction: When retrieval is harder to specify than adaptation. In: Watson, I., Weber, R. (eds.) Case-Based Reasoning Research and Development. pp. 309–324. Springer International Publishing, Cham (2020)
10. Li, J., Li, J., Wang, D., Mao, C., Guan, Z., Liu, Z., Du, M., Qi, Y., Wang, L., Liu, W., Tang, P.: Hierarchical and partitioned planning strategy for closed-loop devices in low-voltage distribution network based on improved kmeans partition method. Energy Reports **9**, 477–485 (2023). https://doi.org/https://doi.org/10.1016/j.egyr.2023.05.161, https://www.sciencedirect.com/science/article/pii/S2352484723009137, 2022 The 3rd International Conference on Power and Electrical Engineering
11. Liang, M., Chang, T., An, B., Duan, X., Du, L., Wang, X., Miao, J., Xu, L., Gao, X., Zhang, L., Li, J., Gao, H.: A stacking ensemble learning framework for genomic prediction. Frontiers in Genetics **12** (2021). https://doi.org/10.3389/fgene.2021.600040, https://www.frontiersin.org/articles/10.3389/fgene.2021.600040

12. Mang, L., Tianpeng, C., Bingxing, A., Xinghai, D., Lili, D., Xiaoqiao, W., Jian, M., Lingyang, X., Xue, G., Lupei, Z., Junya, L., Huijiang, G.: A stacking ensemble learning framework for genomic prediction. Frontiers in Genetics (2021). https://doi.org/10.3389/fgene.2021.600040
13. Petrovic, S., Khussainova, G., Jagannathan, R.: Knowledge-light adaptation approaches in case-based reasoning for radiotherapy treatment planning. Artificial Intelligence in Medicine **68**, 17–28 (2016). https://doi.org/https://doi.org/10.1016/j.artmed.2016.01.006, https://www.sciencedirect.com/science/article/pii/S093336571630015X
14. Rakesh, S., Hegde, N.P., Venu Gopalachari, M., Jayaram, D., Madhu, B., Hameed, M.A., Vankdothu, R., Suresh Kumar, L.: Moving object detection using modified gmm based background subtraction. Measurement: Sensors **30**, 100898 (2023). https://doi.org/https://doi.org/10.1016/j.measen.2023.100898, https://www.sciencedirect.com/science/article/pii/S2665917423002349
15. Richter, M.M., Michael, M.: Knowledge containers. Readings in case-based reasoning. Morgan Kaufmann Publishers (2003)
16. Roldan Reyes, E., Negny, S., Cortes Robles, G., Le Lann, J.: Improvement of online adaptation knowledge acquisition and reuse in case-based reasoning: Application to process engineering design. Engineering Applications of Artificial Intelligence **41**, 1–16 (2015). https://doi.org/https://doi.org/10.1016/j.engappai.2015.01.015, https://www.sciencedirect.com/science/article/pii/S0952197615000263
17. Saxena, N., Kumar, R., Rao, Y.K.S.S., Mondloe, D.S., Dhapekar, N.K., Sharma, A., Yadav, A.S.: Hybrid knn-svm machine learning approach for solar power forecasting. Environmental Challenges **14**, 100838 (2024). https://doi.org/https://doi.org/10.1016/j.envc.2024.100838, https://www.sciencedirect.com/science/article/pii/S2667010024000040
18. Uysal, F., Sonmez, R.: Bootstrap aggregated case-based reasoning method for conceptual cost estimation. Buildings **13**(3) (2023). https://doi.org/10.3390/buildings13030651, https://www.mdpi.com/2075-5309/13/3/651
19. Yu, L., Li, M.: A case-based reasoning driven ensemble learning paradigm for financial distress prediction with missing data. Applied Soft Computing **137**, 110163 (2023). https://doi.org/https://doi.org/10.1016/j.asoc.2023.110163, https://www.sciencedirect.com/science/article/pii/S1568494623001813