An enhanced support vector regression model for agile projects cost estimation

Assia Najm[1], Abdelali Zakrani[2], Abdelaziz Marzak[1]
[1]Department of Mathematics and Computer Sciences, FSB M'sik, Hassan II University, Casablanca, Morocco
[2]Department of Computer Science Engineering, ENSAM, Hassan II University, Casablanca, Morocco

Article Info

Article history:

Received Apr 18, 2021
Revised Dec 16, 2021
Accepted Dec 30, 2021

Keywords:

Agile projects
Optimized artificial immune network
Software cost prediction
Software effort estimation
Support vector regression

ABSTRACT

The appearance of agile software development techniques (ASDT) since 2001 has encouraged many organizations to move to an agile approach. ASDT presents an opportunity for researchers and professionals, but it has many challenges as well. One of the most critical challenges is agile effort prediction. Hence, many studies have investigated agile software development cost estimation (ASDCE). The objective of this study is twofold: First, to propose an improved model based on support vector regression with radial bias function kernel (SVR-RBF) enhanced by the optimized artificial immune network (Optainet). Second, to perform a detailed comparative analysis of the proposed method compared to other existing optimization techniques in the literature and applied for ASDCE. The experimental evaluation was carried out by assessing the performance of the proposed method using some trusted measures like standardized accuracy (SA), mean absolute error (MAE), prediction at level p (Pred(p)), mean balanced relative error (MBRE), mean inverted balanced relative error (MIBRE), and logarithmic standard deviation (LSD). Throughout a dataset with 21 agile projects using the leave-one-out cross-validation (LOOCV) technique. The results obtained prove that the proposed model enhances the accuracy of the SVR-RBF model, and it outperforms the majority of existing models in the literature.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Assia Najm
Department of Mathematics and Computer Sciences, FSB Msik, Hassan II University
Casablanca, Morocco
Email: assia.najm-etu@etu.univh2c.ma

1. INTRODUCTION

In the past twenty years and since the announcement of twelve principles of the agile manifesto in 2001, researchers have shown an increased interest in agile software development. Agile is a generic term that encompasses different methods, approaches, practices, and techniques that meet manifesto principles and values [1]. The agile software development techniques (ASDT) have many attractive advantages like the incremental and iterative aspect of development, quick releases, and good adaptation to the volatile nature of customer requirements. Among the well-known ASDT, we found feature-driven development [2], eXtreme programming [3], and the scrum framework [4]. Nevertheless, for all these ASDT, one of the most severe challenges is to produce an initial reliable and accurate estimate of the project to be developed.

The primary element to estimate the needed development effort is the project size. Principally, story point is a highly employed measurement to deduce the size and the complexity of an agile project. Besides, the velocity factor refers to the number of story points conveyed by the team in a sprint. It significantly impacts the development effort of agile projects. Thus, we used these two measures to predict the effort needed using ASDT.

Generally, cost estimates have an important impact on the process of software development [5]. In fact, accurate estimation leads to the success of projects, while inaccurate predictions lead to the failure of projects [6]. For example, underestimation results in less budget and fewer resources leading to a project with low quality that fails to be accomplished within estimated deadlines. While overestimation leads to expecting more resources than what is needed. Therefore, seeking an efficient technique that produces accurate estimates is useful and of high importance. Especially with the study reported by the standish-group-international (SGI) [7], where more than sixty percent of projects partially or completely fail (19% failed, 44% remains challenged, and only 37% successful).

According to Britto *et al*. [8], various algorithmic and non-algorithmic techniques have been used for agile software cost estimation (ASDCE) purposes. Unfortunately, the accuracy of effort estimations is still a significant challenge. Moreover, it did not achieve acceptable estimation accuracy. This result has been also confirmed by the systematic literature review conducted by Fernandez-Diego *et al*. [9]. In the same study, the authors reported some accuracy improvement; however, many papers present unacceptable estimations. This is why there is still a need to enhance accuracy by proposing new improved ASDCE models or optimizing the existing ones.

Furthermore, prediction techniques couldn’t be assessed without applying suitable accuracy measures. According to [9], the most commonly used measures are based on the median or the mean of magnitude of relative error (MMRE, MdMRE), with 35 studies from 73 and Pred (25%). However, considering the inconvenience of MMRE and its sensitivity to outliers, further measures like the percentage of accuracy, mean absolute error (MAE), and R2 have been used in conjunction to raise the appropriateness of the accuracy. That’s why our second goal from this study is to assess the performance of our proposed model using some trusted criteria as recommended in many papers in the literature like in [10]–[12].

Up to now, some studies have investigated the use of the support vector regression (SVR) model for agile see [13]–[17] thanks to its power in dealing with the agile software effort estimation concerns. However, an optimal SVR model is still needed. In this sense, Zakrani *et al*. [17] have proposed an optimized SVR model using a grid search (GS) method to tune the hyperparameters of the SVR model where every hyperparameter combination is evaluated. This exhaustive search is perceived as inefficient, the time will exponentially grow according to the dimension of evaluations. Also, the population size is user-defined and not adjustable which affects the efficiency of the algorithm and might not find the true optimum.

This present study proposes an improved SVR model with RBF (SVR-RBF) kernel for ASDCE based on the Optainet algorithm. The Optainet [18] is used to tune the SVR-RBF hyperparameters to find the best configuration model that gives accurate estimates. The Optainet explores the effectiveness of the aiNet theory, which simulates the human immune system activity (it has a robust memory and a remarkable ability to differentiate the self cells from the foreign ones). The Optainet algorithm has many advantages regarding the GS method: i) Inclusion of stopping criteria which controls the time of execution, ii) Ability to check and maintain various optimal solutions, iii) Capacity to exploit and explore the whole space of search, and iv) The capability of dynamic adjustment for the population size.

Our major objectives through this study are:

- We suggested an improved SVR-RBF model using the optimized artificial immune network to seek optimal parameters of the SVR-RBF model. This proposed hybrid model enhances the performance of the estimations.
- We applied a very known and trusted empirical evaluation protocol using the following criteria: Pred, LSD, MIBRE, MBRE, and MAE.
- We performed a robust comparative empirical study to assess the suggested model's accuracy compared to the accuracy of other techniques.

To synthesize the goal from this study, we aim during our analysis to answer the following three questions (Qus):

- Qu1: Does the Optainet enhance the performance of the predictions produced using SVR-RBF?
- Qu2: How are the estimations of the SVR-RBF using Optainet in comparison with other optimization methods?
- Qu3: Is the performance of our model more accurate than the other models?

The remainder of this study is organized in: section 2 highlights the previous ASDCE-related works. Section 3 describes the suggested model and the used optimization technique. Section 4 presents the overall experimental evaluation while the obtained results are outlined in section 5, and section 6 reports the conclusion.

2. RELATED WORKS

Multiple papers have addressed ASDCE problems. For example, in [9], 73 relevant studies have been identified in a systematic literature review (SLR) on ASDCE papers published between 2014 and 2020. This SLR presents an update of Britto *et al*. [8] SLR study on ASDCE methods published between 2001 and November 2013 (where 25 papers were selected). According to [8], various techniques have been used for ASDCE purposes, but the most used techniques are those qualified as expert-based prediction techniques. Moreover, most methods did not achieve an acceptable estimation accuracy regarding the observed difference between the real effort value and the predicted one. Notably, the accuracy achieved by the majority of studies did not meet the suggested 25% threshold indicated in [19] for assessing the accuracy of cost prediction. For example, there are commonly accepted thresholds for MMRE and Pred measures, i.e an acceptable MMRE value should be under or equal to 25%, and an acceptable Pred (0.25) should be above or equal to 75%.

In [9], six ASDT were identified: extreme programming (XP), Scrum, feature-driven development, distributed agile software development, Kanban, and agile unified process. Overall, expert-based techniques played a significant role in cost prediction such as the planning poker method, which is intently related to story points sizing metric. Besides, the estimation accuracy challenge, some studies report a suitable value of accuracy, while many papers stated insufficient precision. Also, about 29% of studies used the accuracy measures, while nearly 18% of studies used effect size metric to compare methods. Although ASDT remains dependent on expert judgment (notably, 24.66% of studies used the planning poker method), it is revealed that there has been a significant increase in the number of papers using machine learning techniques, exactly 20.55% of papers [9]. Next, we briefly introduce some examples.

In [20], the authors suggested a model for ASDCE based on user stories, which is adapted to the agile specificity (iterative and adaptive methodology). The empirical evaluation was made based on 21 software agile projects and using MMRE and Pred accuracy measures, and the results show acceptable effort estimations values based on the two metrics. The work in [21], proposed a new formula using story point and velocity variables to estimate the cost. The combination of two optimization techniques, namely particle swarm optimization (PSO) and bee colony, was employed to determine and optimize the formula’s parameters. The results revealed that the proposed technique performs better than the models proposed in [20], [22] based on MAR, R2, Pred (8%), MdMRE, and MMRE.

In [14], the authors attempted to enhance the prediction of ASDT through the use of SVR with several kernels. The model is then evaluated using 21 agile projects. The obtained results demonstrate that SVR-RBF outperforms the other models based on MMRE and Pred (25%) measures. In the same vein, Zakrani *et al*. [17] proposed an improved version of the SVR-RBF model, which uses the grid search optimization method to tune the key hyperparameters. The experimental evaluation was made based on 21 agile projects, and then the model was validated using the LOO-CV method. The results showed that the accuracy of SVR-RBF was improved, and it outperformed other studies in terms of MdMRE, MMRE, and Pred (25%).

Nonetheless, the accuracy reported in the previous studies still needs further enhancement and a robust comparative study based on reliable accuracy measures and a significant statistical test. This is exactly what motivated our present study where we optimize SVR-RBF by Optainet and conduct a comprehensive comparative analysis. The following subsections provide more details about the proposed model.

3. THE PROPOSED METHOD: SVR-RBF-OPTAINET

3.1. SVR model: an overview

The support vector machine (SVM) is a powerful machine learning (ML) model that has offered valuable results in both classification and regression problems [23]. The SVM approach has several attractive features like the sparse way of presenting solutions, good generalization ability, and the capacity to avoid local minimums, thanks to the structural risk minimization concept. In this paper, we employ ε-SVR that introduces the ε-insensitive loss function as shown in Figure 1. It means that all errors less than a defined threshold (inside bars in the figure) are neglected, while errors induced by the points located out of the bars are calculated using ψ and ψ^* like in Figure 1. In the case of nonlinear regression, the following (1) is used where φ denotes a nonlinear function that maps the low input space to the high output space; w represents the weights vector, and c is the threshold.

$$h(x) = w^T\varphi(x) + c \tag{1}$$

We mention that w and c are selected for optimizing the next problem [24]:

Minimizing $_{w,b,\psi,\psi*}$

$$\frac{1}{2} < w, w > \ +B\sum_{i=1}^{l}(\Psi_i + \Psi_i^*)$$

Subjects to

$$(< w, \varphi(x_i) > +c) - \ y_i \leq \ \varepsilon + \psi_i,$$

$$y_i - (< w, \varphi(x_i) > + \, c) \leq \ \varepsilon + \Psi_i^*,$$

$$\psi_i, \Psi_i^*, \geq 0. \qquad (2)$$

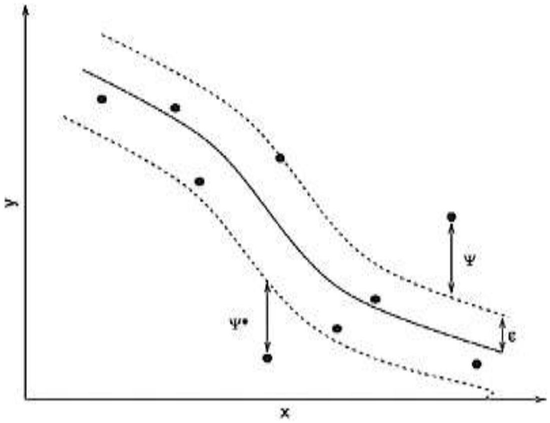

Figure 1. The ε-SVR regression

The ε represents the function's deviation, the regularization parameter B reflects the trade-off between the error's tolerance above ε and the flatness of h, the ψ, ψ* as proposed in [25] presents the slack variables that define the tolerated deviations above the error ε. The base mechanism of the SVR model is minimizing the objective function which takes into account the norm of w and the loss of $\psi_i, \ and \ \Psi_i^*$as expressed in (2). We note that the Lagrangian multiplier is used in the SVR method, it is related to the dot product of φ (x). It can be realized through the kernel function determined as: H(x_i, x_j)=<φ(x_i), φ(x_j)> which prevents the explicit computation of the φ (x). We refer the reader to [24] for more details. In the present study, we used the SVR-RBF towing to its ability to generate good estimates [11], [12]. The computation of the SVR-RBF kernel is done via the expression: H(x_i, x_j)=exp(−λ||x_i–x_j||^2). Hence, parameter λ should be cautiously chosen beside the B and the ε parameters.

3.2. Optainet algorithm

The artificial immune network (aiNet) model was introduced by Castro and Zuben [26]. It is a graph composed of nodes (antibodies) and edges. It is essentially based on the immune system (IS) theory [27] where the IS generates many antibodies or attempts to provide the best-suited antibodies for attacking antigens. The modelization of this phenomenon corresponds to a function optimization process. Many algorithms inspired by the biological domain and especially by the IS were used to achieve global optimization like the opt-IA [28], the B-Cell algorithm [29], the opt-aiNet [18], and the Clonalg algorithm [30].

In this work, the SVR-RBF with Optainet optimization [18] is used to optimize the SVR-RBF parameters. The Optainet explores the effectiveness of the aiNet theory, which simulates the human immune system activity (it has a robust memory and a remarkable ability to differentiate the self cells from the foreign ones). The Optainet algorithm has the following advantages:

- Inclusion of stopping criteria.
- Ability to check and maintain various optimal solutions.

- Capacity to exploit and explore the whole space of search.
- The capability of dynamic adjustment for the population size.

We adopt the following terminology to explain the Optainet algorithm:

- Cells: Each cell in the network is composed of population values. In the Euclidean space, it is denoted with a multivalued vector.
- Cell’s: Fitness stands for the objective function value using a particular network cell.
- Cells: Affinities represent the value of distance (Euclidean) that exists between two cells.
- Cloning: Cloning is producing replication of original cells (or parents).
- Mutation: The generated copies (or offspring) will be mutated to be different from their parents (mutation).

The overall Optainet steps are described as noted below:

BEGIN

a) Randomly initialize the population.

b) If the stopping criteria are not satisfied, do:

- Calculate the fitness of each cell and evaluate it according to the used objective function. Then proceed to its normalization.
- Perform cell cloning according to the chosen Nc, which represents the number of offsprings (or clones).
- Performing the mutation operation, where each clone will be inversely proportional to its parent fitness value as expressed below:

$$cl' = cl + \sigma N(0,1) \tag{3}$$

$$\sigma = (1/\eta)exp(-h^*)$$

where *N(0,1)* presents the random gaussian variable with standard deviation equal to 1 and a zero mean, cl is the parent cell, and the cl’ is the cell resulting from the mutation of cl, also h* ∈[0,1] represents the normalized fitness of a cell, while the parameter η controls the exponential function’s decay. Acceptable mutations are those that are within the domain interval.

- Calculate the fitness of each cell present in the population (including cloned and mutated ones).
- Select the highest fitness cells per clone and exclude the others.
- Evaluating all cell affinities, then suppressing the cells having affinities lower than a predetermined suppression minimum σs (threshold).
- Back into the second step, after adding a p% of random cells.

c) Else, selecting the highest fitness cell and Ending the algorithm.

END

3.3. Optainet based SVR-RBF model

3.3.1. Cell design

We used the ε-SVR model with RBF kernel to implement the proposed model. So, the parameters (B, λ, ε), should be cautiously chosen and carefully optimized using our suggested Optainet-based model. For that reason, each cell in the network includes four elements: the *B*, λ, ε, and features bits. Figure 2 shows the cell design representation. The three first parts represent the SVR-RBF parameters.

B_i
λ_i
ε_{ik}

Figure 2. The cell with three parts, the B, λ, ε

3.3.2. Fitness function

Various criteria were suggested and employed in the literature to evaluate the performance of software cost prediction models. While Pred, MMRE, and MdMRE represent the frequently used ones in many studies [31]. For example, Braga *et al.* [32], and Shin and Goel [33]. Therefore, in this study, we employ a fitness function based on two criteria (4) where we excluded MMRE because it is susceptible to bias. An acceptable value for Pred (25%) is more than or equal to 75%, while for MMRE, MdMRE values must be less than or equal to 0.25. So, we look for cells with higher values of Pred and small values of MdMRE.

$$Fitness\_function = (100 - Pred(25\%)) + MdMRE \tag{4}$$

3.3.3. SVR-RBF-optainet design

We describe in this subsection the architecture of our model for parameter optimization. Especially, the design of the SVR-RBF-Optainet model. Figure 3 shows the main steps of our proposed models.

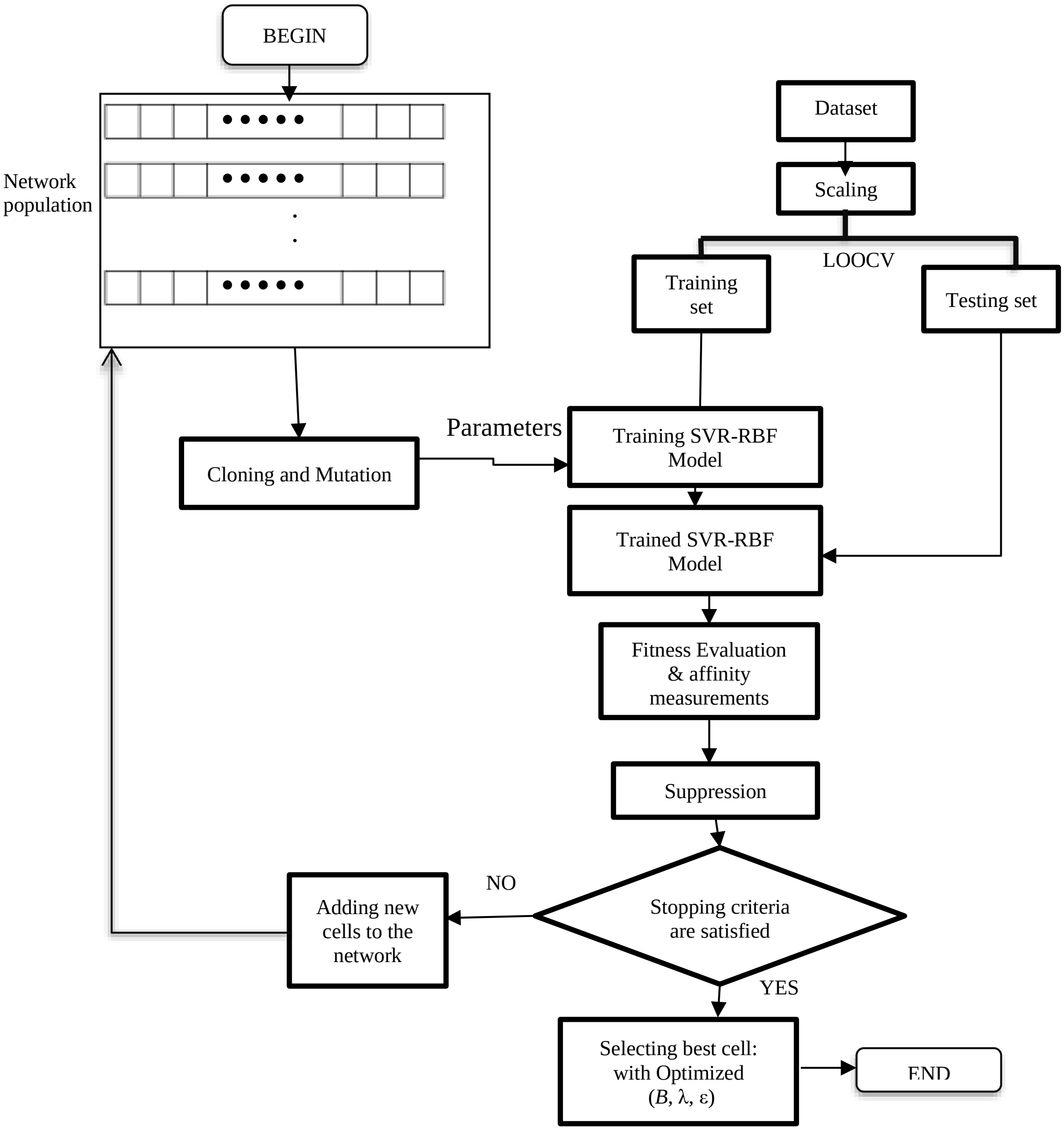

Figure 3. Design of the SVR-RBF-Optainet model

The main steps of our proposed model are described next:

- Data scaling: We firstly proceed to the data-preprocessing phase, where we normalize all the available data in the range between 0 and 1 to strengthen the accuracy of the regressor. The expression used for normalization is noted below:

$$y^* = \frac{y - y_{min}}{y_{max} - y_{min}} \tag{5}$$

where y_{max} is the maximal data value in the dataset, y_{min} is the minimal data value, y is the original value, and y^* is the value after normalization.

- After that, the main steps of the Optainet algorithm were performed: Cloning, mutation, fitness evaluation, affinity measurement, and network suppression as shown in Section 3.2. The output of this algorithm is finding the best solution with parameter optimization.

The Optainet algorithm seeks the optimal SVR-RBF configuration by exploring the parameters search space provided in Table 1. This is done by respecting the fitness function. Hence, the optimal solution that maximizes the fitness function is then chosen.

Table 1. The values of SVR-RBF model and Optainet algorithm

Methods	Hyperparameters

Optainet	Number of iteration={500}
	Size of population={50}
	Number of clones={20}
	The threshold of suppression={0.8}
	Newcomers percentage={50%}

SVR	Type={eps-regression}
	Kernel function={RBF}
	Range values of Hyparameters
	Complexity=[1000, 1500, 3000, 4000, 5000]
	Kernel parameter=[0.9,1,2]
	Epsilon=[0.01, 0.02,0.03]

4. EMPIRICAL DESIGN

4.1. Description of dataset

We employ a dataset that includes twenty-one agile projects with two features: the velocity and the story points. A summarized statistical description is presented in Table 2. It includes the number of features, the dataset size, min, max, mean, median, kurtosis, and skewness of cost. According to kurtosis and skewness values, we notice that the distribution of the cost is not normal besides the weak existence of outliers.

Table 2. Descriptive statistics of the dataset

Elements	Values				Cost		
Dataset size	21	Mini	Maxi	Mean	Median	Skewness	Kurtosis
Features' number	2	21	112	56.43	52	0.6049	-0.7679

4.2. Performance measures and validation method

Concerning the validation method, we employ the LOOCV. The LOOCV is chosen for multiple reasons: it is qualified as a deterministic method, unlike other techniques of validation [34], it produces estimations with large variance and less bias [35]. The LOOCV process is made by excluding one project (test project) from the dataset and performing the estimations with the rest (training set).

Miyazaki *et al.* [36] proposed two accuracy metrics: the mean inverted balanced relative error (MIBRE), and the mean balanced relative error (MBRE) (7) and (8) accordingly. The authors claim that these proposed measures are insensitive to asymmetry and bias contrary to the MMRE criteria (11) which is vulnerable to bias and outliers. Additionally, Shepperd and MacDonell [10] have proposed a novel accuracy criterion, which is the standardized accuracy (SA) (14) which relies on the MAE (13). The SA indicates if the estimated method is better in comparison to randomly guess (PT_0). Therefore, and as reported in [10] a value close to one means that the used method performs better than random guesses. Where MAE_{PT_0} represents the average of a large number of randomly guessing runs. In particular, we used 1000 for this work, MAE_{PT_i} represents the average of the absolute error of a specific estimation method i.

$$AbE_i = |y_i - \widehat{y_i}| \tag{6}$$

$$MIBRE = \frac{1}{N}\sum_{i=1}^{N}\frac{AbE_i}{\max\left(y_i, \widehat{y_i}\right)} \tag{7}$$

$$MBRE = \frac{1}{N}\sum_{i=1}^{N}\frac{AbE_i}{\min\left(y_i, \widehat{y_i}\right)} \tag{8}$$

$$MRE = \frac{AE_i}{y_i} \tag{9}$$

$$Pred(25\%) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1 \; if \; MRE_i \leq 25\% \\ 0 \; otherwise\end{cases} \quad (10)$$

$$MMRE = \frac{1}{N}\sum_{i=0}^{N} MRE \quad (11)$$

$$MdMRE = median(MRE) \quad (12)$$

$$MAE = \frac{1}{N}\sum_{i=0}^{N} AbE_i \quad (13)$$

$$SA = 1 - \frac{MAE_{PT_i}}{\overline{MAE_{PT_0}}} \quad (14)$$

$$LSD = \sqrt{\frac{\sum_{i=1}^{n}(\delta_i + \frac{z^2}{2})^2}{n-1}} \quad (15)$$

$$\delta_i = \ln(y_i) - \ln(\hat{y}_i) \quad (16)$$

$$\Delta = \frac{MAR - \overline{MAR_{PT_0}}}{SPT_0} \quad (17)$$

In this present paper, we used an additional criterion (15): the logarithmic standard deviation (LSD) to assess the performance of estimation methods, where y_i presents the actual cost while $\hat{y}_i$ is the ith project's estimated cost, and z^2 is the estimation related to the δ_i variance (16). Besides, we used the effect size metric (Δ) as expressed in (17) to test if the generated estimates are produced by chance or not. The SPT_0 represents the standard deviation of randomly guessing. The interpretation of the values of (Δ) as explained in [37] is: large when the value is near to 0.8, medium if near to 0.5, and small if Δ is near to 0.2. So, we can deduce that the used technique performs better than the baseline technique if its Δ value exceeds 0.5.

4.3. Statistical test

Even though the employed accuracy measures are helpful to compare different cost prediction models (i.e. checking if one method is better than the other), it remains necessary to perform the statistical test to assess the observed difference, i.e., it is significant or not [38]. We mention that we used the Wilcoxon test; it enables us to evaluate the difference between methods using their absolute error (AbE) (6). The significance level used is 0.05.

5. RESULTS AND DISCUSSION

We discuss in the present section all findings resulting from the assessment of the proposed method and the three compared techniques: the SVR-RBF-GS [17], the regression model of Zia's work [20], the SVR-RBF-Weka which means the SVR-RBF evaluated using default values of the weka tool. First, we tune our model using the Optainet algorithm to get the best model. Second, we evaluate our method and the three aforementioned methods (used in comparison) using Δ and SA metrics. We reported the obtained values in Table 3.

Table 3. The evaluation of each technique based on the SA and the effect size-the highest values are in bold

Approaches	SA	Δ	References
SVR-RBF-Optainet	**0.896**	**6.231**	-
SVR-RBF-GS	0.893	6.138	[17]
Zia *et al*. regression	0.865	5.755	[20]
SVR-RBF-Weka	0.195	1.261	-

A large Δ value means that the assessed technique doesn't produce prediction by chance. Also, a large value of SA means that the evaluated technique generates acceptable estimates. Therefore, in conformity with LOOCV and the values of Table 3, all techniques outperform the random guess (SA>0). Yet, The SVR-RBF-Optainet holds the highest SA value (SA=89.6%), and the SVR-RBF-Weka model holds the lowest one (SA=0.195) despite its better performance against random guessing. All techniques in Table 3 have a large Δ value (Δ >0.5) while SVR-RBF-Optainet retains the largest one (Δ=6.23). Overall, all generated estimates weren’t made by chance.

We note the insufficiency of SA and Δ measures to draw any conclusions concerning the accuracy of our method in comparison to the others. Thus, we applied additional metrics to rank all used techniques throughout the Borda counting method as shown in Table 4. According to Table 4, we recognize the superiority of our method regarding others. The SVR-RBF-Optainet was ranked first, followed by SVR-RBF-GS while the SVR-RBF-Weka can be considered as the worst method for being classed last. In sum, the obtained results reveal that our proposed model, besides all compared methods, outperforms the random guess (SA>0). Also, SVR-RBF-Optainet is the best (ranked first). To make clear and justify the yielded conclusion, the statistical Wilcoxon test was performed. This test was made based on the absolute error AbE and with 0.05 as a significant level. Table 5 shows the resulting p-values. We notice that SVR-RBF-Optainet significantly outperforms the SVR-RBF-Weka. This result confirms its ranking obtained from the Borda count method. However, the difference in performance regarding the others (p-values) is not statistically significant.

Table 4. The ranks of methods based on borda count

Ranks	Techniques	References
1	SVR-RBF-Optainet	-
2	SVR-RBF-GS	[17]
3	Zia *et al.* regression	[20]
4	SVR-RBF-Weka	-

Table 5. The *p*-values of methods based on statistical test.

Methods	SVR-RBF-Weka	Zia *et al.* regression	SVR-RBF-GS
p-values	0.0000	0.1228	0.7525

To deeply analyze the performance of our proposed model, we compare it to many studies reported recently in literature which is the case for the following papers: [14], [17], [20]–[22], [39]. However, these articles applied various performance metrics and validation techniques. For instance, in [21], the authors didn’t mention the technique used for validation, and they calculated the Pred (8%) rather than the frequently employed prediction which is Pred (0.25). Also, in [39], the employed measures are MdMER and MMER rather than MdMRE and MMRE. Even though we attempted to compare our model to the previously cited studies by being consistent with the experimental design which is commonly used in cost prediction as shown in Table 6.

Table 6. The comparison between methods based on the values of MdMRE, MMRE, and Pred

Techniques	Pred (25%)	MdMRE	MMRE	References
SVR-RBF-Optainet (the proposed model)	100	0.0392	0.0596	-
	Pred (0.08)=71.428	0.0397: MdMER	0.0602: MMER	
SVR-RBF-GS	100	0.0426	0.0620	[17]
	Pred (0.08)=66.667	0.0408: MdMER	0,0613: MMER	
SVR-RBF-Weka	28.571	0.4710	0.4671	-
SVR-RBF	95.9052	NA	0.0747	[14]
Zia *et al.* regression model	57.14	0.0714	0.0719	[20]
ABC-PSO	100	0.0333	0.0569	[21]
	Pred (8%)=66.667	0,0344: MdMER	0,0564: MMER	
DT*	38.09	NA	NA	[39]
SGB*	85.71	NA	NA	
RF*	66.67	NA	NA	
GRNN**	85.9182	NA	0.3581	[22]
PNN**	87.6561	NA	1.5776	
GMDH-PNN**	89.6689	NA	0.1563	
CCNN**	94.7649	NA	0.1486	

*DT: Decision Tree, SG: Stochastic Gradient Boosting, RF: Random Forest.
**GRNN: General Regression Neural Network, PNN: Probabilistic Neural Network, GMDH-PNN: GMDH PolynomialNeural Network, CCNN: Cascade-Correlation Neural Network.

We observe from Table 6 that our model performs better than all compared models based on all metrics, apart from the ABC-PSO model in terms of MMRE, and MdMRE (where it has a slightly higher MdMRE (+0.006), and MMRE (+0.002)). Even if our model produces similar Pred (25%) than that of the

ABC-PSO, we observe that it outperforms it greatly based on the value of Pred (8%) with exactly 71.42% for SVR-RBF-Optainet against 66.67% for ABC-PSO) so with a remarkable improvement of (+4.76). We notice that the ABC-PSO method can be excluded from the comparison since the author didn't mention which validation technique was employed. So, they might have used the whole dataset to train the model and test it as well. Finally, from our conducted analysis, we conclude that the Optainet-based SVR-RBF is a promising agile effort estimation model that can produce reliable and accurate estimates.

6. CONCLUSION

The present work proposed an enhanced model for ASDCE purposes. This model is based on the optimized ainet that tunes the parameters of the SVR-RBF model. We assessed our model's performance and made a detailed comparison analysis utilizing the trusted protocol that is based on: (SA, Δ, Borda counting and significance test) and by using the most used criteria, which are the Pred (25%), MdMRE, and MMRE. The findings indicate the superiority of our method regarding others. According to the statistical test results, the SVR-RBF-Optainet was ranked first based on the Borda counting method, and it significantly outperforms the SVR-RBF. Additionally, The SVR-RBF-Optainet outperforms the SVR-RBF regarding the Pred (25%) (+0.42) and MMRE (-0.1). Moreover, our model performs better than all compared models based on all metrics, apart from the ABC-PSO model in terms of MMRE, and MdMRE. In addition, the SVR-RBF-Optainet outperforms the SVR-RBF-GS, and all SVR models using different kernel optimization. Besides, the SVR-RBF-Optainet outperforms greatly Zia's regression based on the MdMRE, MMRE, and Pred. Also, it performs better than the ABC-PSO regression technique in terms of Pred (8%). Finally, we can conclude that the Optainet based SVR-RBF model is a powerful technique. However, we are unable to confirm its superiority in all situations. So, further research should be done using various datasets to generalize the results.

REFERENCES

[1] *Agile practice guide. Newtown square*, Project Ma. Pennsylvania: The Project Management Institute, 2017.

[2] S. R. Palmer and M. Felsing, *A practical guide to feature-driven development*. Pearson Education, 2002.

[3] K. Beck, *Extreme programming explained: embrace change*. 2000.

[4] K. Schwaber, "SCRUM development process," in *Business Object Design and Implementation*, London: Springer London, 1997, pp. 117–134.

[5] A. Trendowicz and R. Jeffery, *Software project effort estimation*. Cham: Springer International Publishing, 2014.

[6] A. ZAKRANI, M. HAIN, and A. IDRI, "Improving software development effort estimating using support vector regression and feature selection," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 4, pp. 399–410, Dec. 2019, doi: 10.11591/ijai.v8.i4.pp399-410.

[7] L. M. Alves, G. Souza, P. Ribeiro, and R. J. Machado, "Longevity of risks in software development projects: a comparative analysis with an academic environment," *Procedia Computer Science*, vol. 181, pp. 827–834, 2021, doi: 10.1016/j.procs.2021.01.236.

[8] R. Britto, V. Freitas, E. Mendes, and M. Usman, "Effort estimation in global software development: a systematic literature review," in *2014 IEEE 9th International Conference on Global Software Engineering*, Aug. 2014, pp. 135–144, doi: 10.1109/ICGSE.2014.11.

[9] M. Fernandez-Diego, E. R. Mendez, F. Gonzalez-Ladron-De-Guevara, S. Abrahao, and E. Insfran, "An update on effort estimation in agile software development: a systematic literature review," *IEEE Access*, vol. 8, pp. 166768–166800, 2020, doi: 10.1109/ACCESS.2020.3021664.

[10] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Information and Software Technology*, vol. 54, no. 8, pp. 820–827, Aug. 2012, doi: 10.1016/j.infsof.2011.12.008.

[11] M. Hosni, A. Idri, A. Abran, and A. B. Nassif, "On the value of parameter tuning in heterogeneous ensembles effort estimation," *Soft Computing*, vol. 22, no. 18, pp. 5977–6010, Sep. 2018, doi: 10.1007/s00500-017-2945-4.

[12] A. Idri and I. Abnane, "Fuzzy analogy based effort estimation: an empirical comparative study," in *2017 IEEE International Conference on Computer and Information Technology (CIT)*, Aug. 2017, pp. 114–121, doi: 10.1109/CIT.2017.29.

[13] E. Scott and D. Pfahl, "Using developers' features to estimate story points," in *Proceedings of the 2018 International Conference on Software and System Process*, May 2018, pp. 106–110, doi: 10.1145/3202710.3203160.

[14] S. M. Satapathy, A. Panda, and S. K. Rath, "Story point approach based agile software effort estimation using various SVR kernel methods," in *The 26th International Conference on Software Engineering and Knowledge Engineering (SEKE 2014)*, 2014, pp. 304–307.

[15] S. Porru, A. Murgia, S. Demeyer, M. Marchesi, and R. Tonelli, "Estimating story points from issue reports," in *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering*, Sep. 2016, pp. 1–10, doi: 10.1145/2972958.2972959.

[16] O. Malgonde and K. Chari, "An ensemble-based model for predicting agile software development effort," *Empirical Software Engineering*, vol. 24, no. 2, pp. 1017–1055, Apr. 2019, doi: 10.1007/s10664-018-9647-0.

[17] A. Zakrani, A. Najm, and A. Marzak, "Support vector regression based on grid-search method for agile software effort prediction," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Oct. 2018, pp. 1–6, doi: 10.1109/CIST.2018.8596370.

[18] L. N. de Castro and J. Timmis, "An artificial immune network for multimodal function optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 1, pp. 699–704, doi: 10.1109/CEC.2002.1007011.

[19] S. D. Conte, H. E. Dunsmore, and V. Y. Shen, *Software engineering metrics and models*. Benjamin-Cummings Publishing Co.,

Inc.Subs. of Addison-Wesley Longman Publ. Co390 Bridge Pkwy. Redwood City, CAUnited States, 1986.

[20] Ziauddin, S. K. Tipu, and S. Zia, “An effort estimation model for agile software development,” *Advances in Computer Science and its Applications (ACSA)*, vol. 2, no. 1, pp. 314–324, 2012.

[21] T. T. Khuat and M. H. Le, “A novel hybrid ABC-PSO algorithm for effort estimation of software projects using agile methodologies,” *Journal of Intelligent Systems*, vol. 27, no. 3, pp. 489–506, Jul. 2018, doi: 10.1515/jisys-2016-0294.

[22] A. Panda, S. M. Satapathy, and S. K. Rath, “Empirical validation of neural network models for agile software effort estimation based on story points,” *Procedia Computer Science*, vol. 57, pp. 772–781, 2015, doi: 10.1016/j.procs.2015.07.474.

[23] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY: Springer New York, 2000.

[24] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004, doi: 10.1023/B:STCO.0000035301.49549.88.

[25] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[26] L. Nunes de Castro and F. J. V. Zuben, “aiNet,” in *Data Mining*, IGI Global, 2002, pp. 231–260.

[27] N. K. Jerne, “Towards a network theory of the immune system,” *Annales d’immunologie*, vol. 125C, no. 1–2, pp. 373–389, Jan. 1974, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/4142565.

[28] V. Cutello, G. Nicosia, and M. Pavone, “Exploring the capability of immune algorithms: a characterization of hypermutation operators,” in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2004, pp. 263–276.

[29] J. Kelsey and J. Timmis, “Immune inspired somatic contiguous hypermutation for function optimisation,” in *Genetic and Evolutionary Computation {\textemdash} {GECCO} 2003*, Springer Berlin Heidelberg, 2003, pp. 207–218.

[30] L. N. de Castro and F. J. Von Zuben, “The clonal selection algorithm with engineering applications,” 2000.

[31] C. J. Burgess and M. Lefley, “Can genetic programming improve software effort estimation? A comparative evaluation,” *Information and Software Technology*, vol. 43, no. 14, pp. 863–873, Dec. 2001, doi: 10.1016/S0950-5849(01)00192-6.

[32] P. L. Braga, A. L. I. Oliveira, G. H. T. Ribeiro, and S. R. L. Meira, “Bagging predictors for estimation of software project effort,” in *2007 International Joint Conference on Neural Networks*, Aug. 2007, pp. 1595–1600, doi: 10.1109/IJCNN.2007.4371196.

[33] M. Shin and A. L. Goel, “Empirical data modeling in software engineering using radial basis functions,” *IEEE Transactions on Software Engineering*, vol. 26, no. 6, pp. 567–576, Jun. 2000, doi: 10.1109/32.852743.

[34] E. Kocaguneli and T. Menzies, “Software effort models should be assessed via leave-one-out validation,” *Journal of Systems and Software*, vol. 86, no. 7, pp. 1879–1890, Jul. 2013, doi: 10.1016/j.jss.2013.02.053.

[35] E. Kocaguneli, T. Menzies, and J. W. Keung, “On the value of ensemble effort estimation,” *IEEE Transactions on Software Engineering*, vol. 38, no. 6, pp. 1403–1416, Nov. 2012, doi: 10.1109/TSE.2011.111.

[36] Y. Miyazaki, A. Takanou, H. Nozaki, N. Nakagawa, and K. Okada, “Method to estimate parameter values in software prediction models,” *Information and Software Technology*, vol. 33, no. 3, pp. 239–243, Apr. 1991, doi: 10.1016/0950-5849(91)90139-3.

[37] J. Cohen, “A power primer,” *Psychological Bulletin*, vol. 112, no. 1, pp. 155–159, 1992, doi: 10.1037/0033-2909.112.1.155.

[38] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[39] S. M. Satapathy and S. K. Rath, “Empirical assessment of machine learning models for agile software development effort estimation using story points,” *Innovations in Systems and Software Engineering*, vol. 13, no. 2–3, pp. 191–200, Sep. 2017, doi: 10.1007/s11334-017-0288-z.

BIOGRAPHIES OF AUTHORS

Assia Najm is a Ph.D. student at Hassan II university at Casablanca, department of Mathematics and Computer Sciences. She received an engineer's degree in telecommunication and technology of information from INPT, Rabat, Morocco, in 2014. Her research interests include software cost estimation, software metrics, fuzzy logic, decision trees, and optimization. She can be contacted at email: assia.najm-etu@etu.univh2c.ma.

Dr. Abdelali Zakrani is an associate professor at Hassan II university at Casablanca, He received the B.Sc. degree in Computer Science from Hassan II University, Casablanca, Morocco, in 2003, and his DESA degree (M.Sc.) and Ph.D. in the same major from University Mohammed V, Rabat, in 2005 and 2012 respectively. His research interests include software cost estimation, software metrics, fuzzy logic, neural networks, decision trees. He can be contacted at email: zakrani@gmail.com

Dr. Abdelaziz Marzak is currently a professor of computer science at the Faculty of Science Ben M’sik Casablanca, where he is the head of technologies of information and communication. He has several publications on the Web and databases. He received a Ph.D. in computer science in 2000 and a Ph.D. in computer science in 1997 both from Mohammed V University at Rabat, Morocco. He can be contacted at email: marzak@hotmail.com