Soft Computing (2019) 23:10881–10900
https://doi.org/10.1007/s00500-018-3639-2

Software effort estimation using FAHP and weighted kernel LSSVM machine

Sumeet Kaur Sehra[1,2] · Yadwinder Singh Brar[1] · Navdeep Kaur[3] · Sukhjit Singh Sehra[4]

Published online: 4 December 2018

Abstract
In the life cycle of software product development, the software effort estimation (SEE) has always been a critical activity. The researchers have proposed numerous estimation methods since the inception of software engineering as a research area. The diversity of estimation approaches is very high and increasing, but it has been interpreted that no single technique performs consistently for each project and environment. Multi-criteria decision-making (MCDM) approach generates more credible estimates, which is subjected to expert's experience. In this paper, a hybrid model has been developed to combine MCDM (for handling uncertainty) and machine learning algorithm (for handling imprecision) approach to predict the effort more accurately. Fuzzy analytic hierarchy process (FAHP) has been used effectively for feature ranking. Ranks generated from FAHP have been integrated into weighted kernel least square support vector machine for effort estimation. The model developed has been empirically validated on data repositories available for SEE. The combination of weights generated by FAHP and the radial basis function (RBF) kernel has resulted in more accurate effort estimates in comparison with bee colony optimisation and basic RBF kernel-based model.

Keywords Software effort estimation · Fuzzy analytic hierarchy process · Least square support vector machine

1 Introduction

Software engineering (SE) discipline has evolved since the 1960s and has garnered significant knowledge (Zelkowitz et al. 1984). Academia and industry have invested in SE research and development in past decades that resulted in the development of improved tools, methodologies and techniques. Over the years, there has been an intense criticism of SE research as it advocates more than it evaluates (Glass et al. 2002). Many researchers have attempted to characterise SE research, but they failed to present a comprehensive picture (Jørgensen et al. 2009; Shaw 2002).

Software effort estimation (SEE) is a critical component that predicts the effort to accomplish development or maintenance tasks based on historical data. Accurate estimates are critical to company and customers because it can help the company personnel to classify, prioritise and determine resources to be committed to the project (Nisar et al. 2008). Since its inception, the problems and issues in SEE have been addressed by researchers and practitioners. The researchers have proposed numerous estimation methods since the inception of SE as a research area (Jørgensen and Shepperd 2007; Rastogi et al. 2014; Trendowicz et al. 2008). The application of developed models has been found to be appropriate for the specific types of the development environment. The advances in the technology stack and frequent changing user requirements have made the process of SEE difficult. Numerous approaches have been tried to predict this probabilistic process accurately, but no single technique has performed consistently. Even, few researchers have tried to employ a combination of the approaches rather than a single approach. The major reason for inaccurate estimates is that datasets of past projects are usually sparse, incomplete, inconsistent and not well documented. Another reason for this is that SEE process is dependent on multiple seen and unseen factors.

Communicated by V. Loia.

✉ Sumeet Kaur Sehra
sumeetksehra@gmail.com

[1] I.K.G. Punjab Technical University, Jalandhar, Punjab, India

[2] Guru Nanak Dev Engineering College, Ludhiana, Punjab, India

[3] Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India

[4] Elocity Technology Inc., Toronto, Canada

Despite extensive research on it, the community is unable to develop and accept a single model that can be applied in diverse environments and which can handle multiple environmental factors. Recently, multi-criteria decision-making (MCDM) methods have emerged as well-qualified approaches to handle multi-factored decision making. Also, incorporation of machine learning (ML) approaches to the primary estimation model has always enhanced the performance. Thus, current paper is motivated to focus on the development of hybrid model based on ML and MCDM. Fuzzy analytic hierarchy process (FAHP) has been used effectively for feature ranking. Ranks generated from FAHP have been integrated into weighted kernel least square support vector machine (LSSVM) for effort estimation. The model developed has been empirically validated on data repositories available for SEE.

The paper has been divided into six sections. The following section describes related work in detail. The third section elaborates the methodology followed. Proposed hybrid model is presented in fourth section. The fifth section provides the empirical validation of the proposed model. The final section concludes the paper and provides directions for future work.

2 Related work

Algorithmic SEE models have been studied for many years (Jørgensen and Shepperd 2007). The application of ML techniques to predict effort has gained significant momentum from the year 2006. The most explored algorithmic approaches such as “fuzzy logic” (Ryder 1998; Wong et al. 2009), “neural networks” (Attarzadeh and Ow 2009; Azzeh and Nassif 2016; Dasheng and Shenglan 2012; Idri et al. 2006, 2002; Sheta et al. 2015) and genetic algorithms (Gharehchopogh et al. 2015; Milios et al. 2013; Oliveira et al. 2010) have been consistently used in every aspect of SEE. Other areas identified are “nature inspired algorithms” which focus on the use of algorithms based upon various natural phenomena (Chalotra et al. 2015a, b; Dave and Dutta 2011; Gharehchopogh et al. 2015; Idri et al. 2007; Madheswaran and Sivakumar 2014; Reddy et al. 2010). “feature selection in problem domain” (Kocaguneli et al. 2015; Liu et al. 2014), “support vector regression” (Braga et al. 2007; Corazza et al. 2011) and “case-based reasoning” (Mendes et al. 2002).

Wen et al. (2012) investigated 84 primary studies of ML techniques in SEE for finding out different ML techniques, their estimation accuracy, the comparison between different models and estimation contexts. Based upon this study, they found that in SEE, eight types of ML techniques have been applied and concluded that ML models provide more accurate estimates as compared to non-ML models.

The effectiveness of support vector regression (SVR) for web effort estimation using a cross-company dataset has been investigated by Corazza et al. (2011). The analysis has showed that different preprocessing strategies and kernels can significantly affect the prediction effectiveness of the SVR approach. It has also revealed that logarithmic preprocessing in combination with the radial basis function (RBF) kernel provides the best results. The estimation of software effort has been performed by using ML techniques instead of subjective and time-consuming estimation methods (Hidmi et al. 2017). Models using two ML techniques, viz. support vector machine (SVM) and k-nearest neighbour (k-NN) separately and combining those together using ensemble learning, have been proposed.

SVM has been widely used in classification and nonlinear function estimation. However, the major drawback of SVM is its higher computational burden for the constrained optimisation programming. This disadvantage has been overcome by LSSVM, which solve linear equations instead of a quadratic programming problem. LSSVM has been employed in numerous applications including stock market trend prediction (Marković et al. 2015), project risk forecasting (Liyi et al. 2010) and analogy-based estimation (ABE) (Benala and Bandarupalli 2016).

There may be numerous factors affecting software effort estimates, but few dominate the given environment of software development, and those must be identified. Effort estimation can be understood as a problem that depends upon multiple factors which are qualitative as well as quantitative.

Minku and Yao (2013) argued that SEE can be considered as a multi-objective problem. Ferrucci et al. (2011) studied the efficacy of multi-criteria genetic programming on effort estimation and stated that effort estimation is an inherently multi-criteria problem. Jiang and Naudé (2007) found project size as the crucial factor in effort estimation.

Shepperd and Cartwright (2001) surveyed a company and discovered that project managers rely on the parameters namely count of programs, functionality, difficulty level, personnel skill and sameness as past work for effort estimation in a particular project. Morgenshtern et al. (2007) considered the factors in mainly four dimensions, namely project uncertainty, use of estimation development, estimation management and the experience of the estimator. Furulund and Molokken-Ostvold (2007) highlighted and confirmed the importance of using experienced-based data and checklists. Jiang and Naudé (2007) considered factors for making a decision, viz. project size, average team size, development language, computer-aided software engineering (CASE) tools, development type and computer platform. They have also discussed that use of historical data can also be used as a mean to increase SEE accuracy.

Liu et al. (2017) suggested new method based on LSSVM and K-means clustering for ranking the optimal solutions for the multi-objective allocation of water resources. An approach has been proposed using feature ranking and feature selection approach in combination with weighted kernel LSSVMs. The feature weights obtained by the analytic hierarchy process (AHP) method are used for feature ranking and selection and used with the LSSVMs through a weighted kernel (Marković et al. 2017).

3 Methodology

3.1 Multi-criteria decision-making (MCDM)

MCDM approaches qualify for SEE as they can combine the historical data and expert judgement by quantifying subjectivity in judgement. The MCDM method is the process of making decisions in the presence of multiple criteria or objectives. The criteria can be multiple and quantifiable or non-quantifiable from which an expert is required to choose. The solution of the problem is dependent on the inclination of the expert since the objectives usually are contradictory (Belton and Stewart 2002). Further, the difficulty of developing a selection criterion for precisely describing the preference of one alternative over another is also a concern. MCDM models include preference ranking organisation method for enrichment evaluation (PROMOTHE), technique for the order of prioritisation by similarity to ideal solution (TOPSIS), AHP, elimination and choice expressing reality (ELECTRE), VIKOR, each of which has a different algorithm to solve the problems (Lee and Tu 2011).

AHP (Menzies et al. 2006) is the most explored MCDM technique having characteristics of both model-based systems and expert judgement. AHP, developed by Saaty (2004), is an MCDM technique which has been used in vast areas. Further, it has also been identified as one of the prime approaches that can be used for supporting the software estimation planning. AHP has the capability of combining historical data and expert judgement by quantifying subjective judgement. The result of this method is to provide a formal and systematic method of extracting, combining and capturing expert judgements and their relationship to similar reference data (Menzies et al. 2006). Despite being a popular approach, there are certain issues which need to be addressed. Firstly, as judgements given by experts are relative, so any arbitrary change in the value of alternatives may affect the weights of other alternatives resulting in a problem known as rank reversal (Wang and Luo 2009). Another issue with AHP is its subjectivity and imprecision due to Saaty's nine-point scale (Saaty 2008). These issues can be handled by adding fuzziness to AHP, thus resulting in new approach called fuzzy analytic hierarchy process (FAHP).

3.1.1 Fuzzy analytic hierarchy process (FAHP)

Fuzzy logic is a well-established approach for handling the subjectivity of human judgements and vagueness of the data (Zadeh 1988). The combination of fuzzy logic and AHP is a hybrid approach for both qualitative and quantitative criteria comparison using expert judgements to find weights and relative rankings. Since most of multi-criteria methods suffer from vagueness, FAHP approach can better tolerate this vagueness as experts are always more confident while using the intervals for estimates rather than fixed values (Mikhailov and Tsvetinov 2004). The combination of fuzzy logic and structural analysis generates more credible results than conventional AHP (Liao 2011; Tang et al. 2005).

In FAHP, expert judgement is represented as a range of values instead of single crisp values (Kuswandari 2004). The range values can be given as optimistic, pessimistic or moderate. Triangular fuzzy number (TFN) is represented by Eq. (1). Here l, m, u are pessimistic, moderate and optimistic values, respectively. The difference between $u - l$ describes the degree of fuzziness of judgement.

$$a_{ij} = (l_{ij}, m_{ij}, u_{ij}) \tag{1}$$

Algorithm 1: Algorithm for FAHP.

```
   Input: Comparison matrix, Fuzzy triangular number for finding
          the relative importance of criteria, Linguistic fuzzy scale.
   Output: Weight matrix, Optimal criterion
   Result: Relative ranking of criteria using weight matrix.
 1 Initialisation:
 2 Decomposing a problem into decision hierarchy consisting of
   interrelated decision elements, in which goal is at the top level
   and lower levels consist of criteria and sub-criteria.
 3 Load the linguistic scale for the fuzzy triangular number.
 4 Perform the relative importance of criteria.
 5 Compute the pairwise comparison matrix.
 6 Compute the consistency of the matrix.
 7 if Consistency ratio (CR) $\geq$ 0.1 then
 8     Go to line 4
 9 else
10     Compute the synthetic extent values.
11     Compute the degree of possibility for fuzzy numbers.
12     Compute the weight matrix for criteria.
13     Compute the normalised weighted decision matrix.
14     Compute the optimal solution.
15 end
16 return
```

There are various methods available to calculate the weights and prioritise ranking of the alternatives (Buckley 1985; Chang 1992, 1996; Csutora and Buckley 2001; Kahraman et al. 2003; Van Laarhoven and Pedrycz 1983). Naghadehi et al. (2009) discussed these methods and recommended for using the method suggested by Chang (1996). Thus, in this

Table 1 Linguistic scale for FAHP

Linguistic scale for importance	Fuzzy numbers for FAHP	Membership function	Domain	Triangular fuzzy scale (l, m, u)
Just equal	1			(1.0, 1.0, 1.0)
Equally important	1	$\mu M(x) = (3 - x)/(3 - 1)$	$1\leq x \leq 3$	(1.0, 1.0, 3.0)
Weak importance over each another	3	$\mu M(x)= (x - 1)/(3 - 1)$	$1 \leq x \leq 3$	(1.0, 3.0, 5.0)
		$\mu M(x) = (5 - x)/(5 - 3)$	$3 \leq x \leq 5$	
Essential importance over each other	5	$\mu M(x) = (x - 3)/(5 - 3)$	$3 \leq x \leq 5$	(3.0, 5.0, 7.0)
		$\mu M(x) = (7 - x)/(7 - 5)$	$5 \leq x \leq 7$	
Very strong importance over other	7	$\mu M(x) = (x - 5)/(7 - 5)$	$5 \leq x \leq 7$	(5.0, 7.0, 9.0)
		$\mu M(x) = (9 - x)/(9 - 7)$	$7 \leq x \leq 9$	
Extreme importance over other	9	$\mu M(x) = (x - 7)/(9 - 7)$	$7 \leq x \leq 9$	(7.0, 9.0, 9.0)

The value of second element in comparison with first would be by reciprocal of TFN given as (1/u1,1/m1,1/l1)

paper for finding the weights, the synthetic extent analysis method given by Chang (1996) has been utilised. The objective of this method is to perform pairwise comparisons. TFN provides an opportunity in deciding the weight of one alternative over the other.

Algorithm 1 describes the steps applied in FAHP. The membership function used for creating the fuzzy set is given in Eq. (2), where x is the weight of relative importance of one criterion over another criterion.

$$\mu_A(x) = \left\{\begin{matrix} 0 \quad \text{for} \quad x \leq l \\ \frac{x-l}{m-l} \quad \text{for} \quad l \leq x \leq m \\ \frac{u-x}{u-m} \quad \text{for} \quad m \leq x \leq u \\ 0 \quad \text{for} \quad x \geq n \end{matrix}\right\} \tag{2}$$

The modified Saaty's scale using TFN (presented in Table 1) has been used to construct the FAHP comparison matrices. The reciprocal property for the TFN is generated as $a_{ji} = (1/u_{ji}, 1/m_{ji}, 1/l_{ji})$. The next step is to use extent analysis method to calculate the relative ranking of alternatives, and the synthetic extent values are obtained by Eq. (3).

$$S_i = \sum_{j=1}^{m} N_{ci}^{j} \otimes \left[\sum_{i=1}^{n} \sum_{j=1}^{m} N_{ci}^{j} \right]^{-1} \tag{3}$$

The degree of possibility of $N_1 \geq N_2$ is defined in Eq. (4).

$$V\,(N_1 \geq N_2) = \sup[\min\,(\mu_{N1}\,(x))\,,\mu_{N2}\,(y))] \tag{4}$$

$$\begin{aligned} V\,(N2 \geq N1) &= hgt\,(N_1 \cap N_2) = \mu_{N1}(d) \\ &= \begin{pmatrix} 1 \quad if \quad m2 > m1 \\ 0 \quad if \quad l1 \geq u2 \\ \frac{l_1-u_2}{(m_2-l_2)-(m_1-l_1)}, \text{otherwise} \end{pmatrix} \end{aligned} \tag{5}$$

In Eq. (5), d is representing ordinate of the highest intersection point between μ_{N1} and μ_{N2}. The degree of possibility for a convex fuzzy number is defined by Eq. (6).

$$\begin{aligned} &V\,(N \geq N_1, N_2, ..., N_k) \\ &\quad = [(N \geq N_1)\,, ..., (N \;\geq N_k)] = \min V\,(N \geq N_i) \end{aligned} \tag{6}$$

In order to normalise the weight vector, Eq. (7) is used.

$$W_A = \frac{W^T}{\sum(W^T)} \tag{7}$$

After calculating the weights of criteria, the scores of alternatives with respect to each criterion are evaluated and composite weights of the decision alternatives are determined by aggregating the weights through the hierarchy.

In a situation, where many pairwise comparisons are performed, inconsistencies may typically arise. The effective method for identifying the inconsistency is by computing consistency index (CI). It refers to the average of the remaining solutions of the characteristic equation for the inconsistent matrix A. This index increases in proportion to the inconsistency of the estimates. The most often approach used for calculating CI is given in Eq. (8).

$$\text{CI} = \frac{\lambda_{\max} - n}{n - 1} \tag{8}$$

where $\lambda_{\max}$ denotes the maximal eigenvalue of the matrix and n is the number of factors in the judgement matrix. When the matrix is consistent, then $\lambda_{\max} = n$ and CI = 0.

CR is a measure of how a given matrix compares to a purely random matrix in terms of their consistency indices. Equation 9 is used for calculating CR.

$$\text{CR} = \frac{\text{CI}}{\text{RI}} \tag{9}$$

Table 2 Random index

N	3	4	5	6	7	8	9	10	11	12
RI(n)	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.54

Here, random index (RI) is the average CI of 100 randomly generated (inconsistent) pairwise comparison matrices. These values are tabulated in Table 2 for different values of n. CR of less than 0.1 (“10% of average inconsistency”) of randomly generated pairwise comparisons (matrices) is usually acceptable. If CR is not acceptable, judgements should be revised.

This has been witnessed that the ranking in FAHP is still based on the decisions of the expert. Further, from the literature, it has been witnessed that ML algorithm provides promising results for labelled data. Thus, there is a need of a hybrid model that can handle knowledge from past dataset, while removing uncertainty in recall issue of the expert and utilise judgement of the expert. To address these aspects, the current research has proposed to merge the MCDM (for handling uncertainty) and ML algorithm (for handling imprecision), to create a robust hybrid method. The rankings of the projects have been calculated using FAHP, and these weights are further used to modify the weight of kernel. The modified kernel has been utilised by LSSVM algorithm for prediction. In this work, RBF kernel has been applied, whereas the generated set of weights from FAHP can also be independently incorporated into other kernel-based functions.

3.2 Support vector machine (SVM)

SVM is a discriminative classifier formally defined by a separating hyperplane as presented in Fig. 1. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorises the new examples. The hyperplane gives the largest minimum distance to the training examples. Therefore, the optimal separating hyperplane maximises the margin of the training data.

SVM is capable of handling and mapping nonlinear classification using kernel trick. The input data are mapped to n-dimensional feature space using function, and then, the linear model is applied in that space, as shown in Fig. 2. Equation (10) describes the mapping function.

$$y = f(x) = \text{sign}[w^T \phi(x) + b] \tag{10}$$

Here, w is the weight vector in the input space, ϕ is a nonlinear function providing the mapping from the input space to a higher (possibly infinite)-dimensional feature space, and b is a bias parameter. For data classification, it is assumed that $y_k[w^T \phi(x_k) + b] \geq,\ k = 1....N$. The function $f(x_k)$ is determined by solving the minimisation of Eq. (11), subject to $y_k[w^T \phi(x_k) + b] \geq\ 1 - \epsilon_k,\ k = 1....N$, where $\epsilon_k \geq\ 0, k = 1....N.$

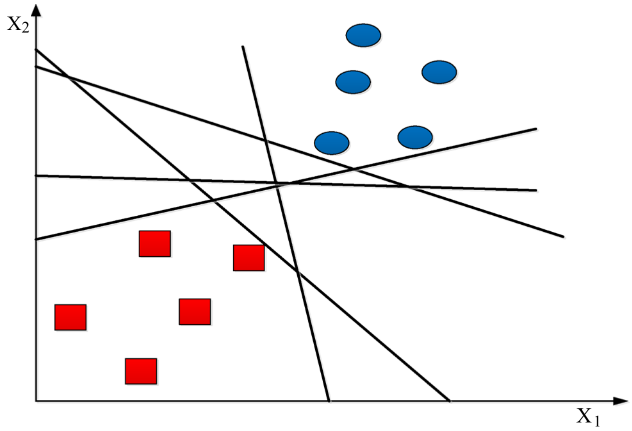


Fig. 1 SVM as a binary classifier

$$\min\ J_p(w, \epsilon) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \epsilon_k \tag{11}$$

Based on this, the Lagrangian dual of the problem is presented as in Eq. (13), which is a quadratic maximisation problem and hence requires the application of quadratic programming methods. Thus, the quadratic maximisation problem is given as in Eq. (13), subject to $\sum_{k=1}^{N} \alpha_k y_k = 0$ and $0 \leq \alpha_k\ \leq c, k = 1, ..., N.$

$$\begin{aligned} L(w, b, \varepsilon, \alpha, v) = J(w, \varepsilon_k) - \sum_{k=1}^{N} \alpha_k y_k [w^T \phi(x_k) + b] \\ -1 + \varepsilon_k - \sum_{k=1}^{N} v_k \varepsilon_k \end{aligned} \tag{12}$$

$$\max\ Q(\alpha) = -\frac{1}{2} \sum_{k,l}^{N} y_k y_l K(x_k, x_l) \alpha_k \alpha_l\ + \sum_{k=1}^{N} \alpha_k \tag{13}$$

Here, $K(x_k, x_l)$ kernel function acts as a dot product to map the input data into a higher-dimensional space, given as $K(x_k, x_l) = \phi(x_k)^T \phi(x_l)$. A detailed explanation of SVM is given by Vapnik (2013).

3.3 Least squares support vector machines (LSSVMs)

LSSVMs are least squares versions of SVM, which are a set of related supervised learning methods that analyse data

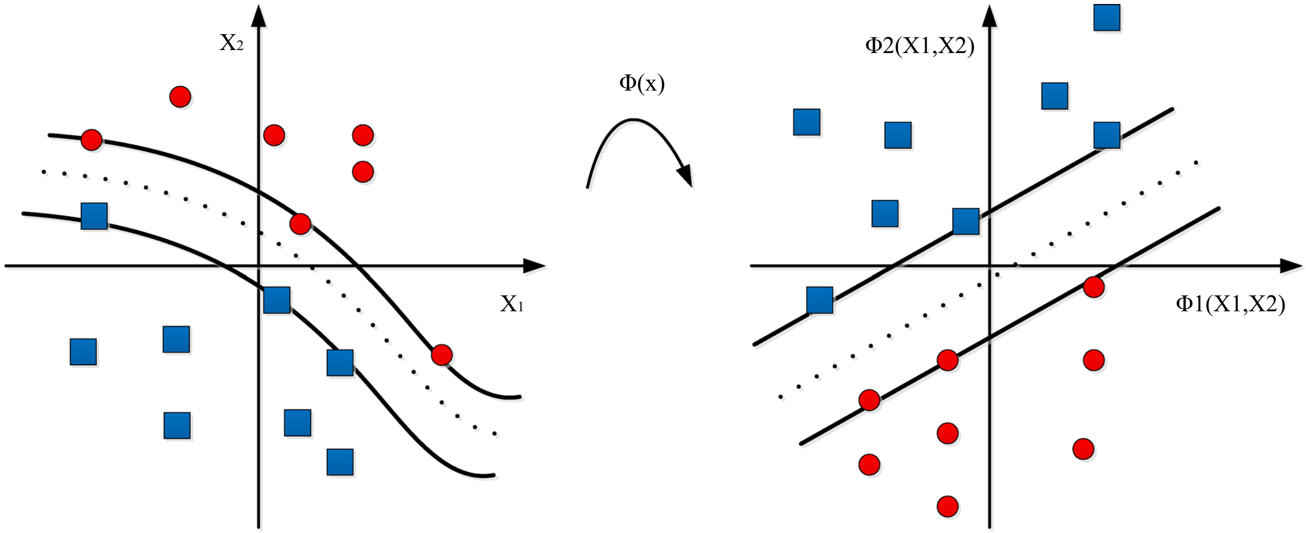


Fig. 2 Conversion of nonlinear data to linear in n-dimensional space

and recognise patterns, and which are used for classification and regression analysis. Another synonym of standard SVM is LSSVM algorithm. It adopts equality constraints and a linear Karush–Kuhn–Tucker system, which has a more powerful computational ability in solving the nonlinear and small-sample problem using linear programming methods (Suykens and Vandewalle 1999). However, the modelling accuracy of a single LSSVM is not only influenced by the input data source but also affected by its kernel function and regularization parameters. LSSVM is a class of kernel-based learning methods. LSSVMs have been effectively used for estimation and nonlinear classification problems. Gestel Suykens et al. (2004) stated that standard SVM and LSSVM perform consistently in a combination of tuned hyperparameters. But, the advantage of using LSSVM is 1) less computational burden for the constrained optimisation programming and 2) better for higher-dimensional data.

In this paper, LSSVM has been utilised to predict the effort of software data, while learning from the past available data of software projects. The function $f(x_k)$ is determined by solving the minimisation of Eq. (14), subject to $1 - e_k = \gamma_k[w^T\phi(x_k) + b],\ k = 1....N.$

$$\min\ J_p(w, e) = \frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 \tag{14}$$

A trade-off is made between the minimisation of the squared error, e_k^2, and minimising the dot product of the weight vectors, $w^T w$, by an optimisation parameter γ. Based on this, the Lagrangian of the problem is presented as in Eq. (15), where α_k is a Lagrangian multiplier.

$$\frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N}\alpha_k(\gamma_k[w^T\phi(x_k) + b] - 1 + e_k) \tag{15}$$

The weighted kernel function is defined as $K(\theta x_k, \theta x_l)$ where θ represents a weight vector comprising of dataset features. This weighted kernel function makes the LSSVM as weighted LSSVM. The classification model of weighted LSSVM is represented as in Eq. (16).

$$y(x) = \text{sign}\left[\sum_{k=1}^{N}\alpha_k y_k K(\theta x_k, \theta x_l) + b\right] \tag{16}$$

At this point, the kernel function is applied which computes the dot product in the higher-dimensional feature space by using the original attribute set. Some of the kernel functions are listed below:

- Linear: $K(x_k, x_l) = {x_l}^T x_k + c$
- Polynomial : $K(x_k, x_l) = (\gamma {x_l}^T x_k + r)^d > 0$
- Multilayer perceptron: $K(x_k, x_l) = \tan\ h(\gamma x_l^T x_k + r)$
- Radial basis function: $K(x_k, x_l) = \exp(-\frac{||x_k - x_l||^2}{\sigma^2})$

Here γ, σ, r and d are kernel parameters. Kernel-based estimation techniques, such as SVM and LSSVM, have shown to be powerful nonlinear classification and regression methods. In this study, RBF kernel has been used since it is previously found to be a good choice in case of LSSVM (Gestel Suykens et al. 2004).

4 Proposed model

A hybrid model has been developed by combining MCDM approach FAHP and ML approach LSSVM. FAHP has been used to generate feature weights, and a weighted kernel (W_k) has been used to assimilate generated weights into the LSSVM. Algorithm 2 describes the steps followed to develop the model.

In this model, the criteria chosen have been effort and lines of code (LOC). The alternatives chosen have been the projects. The first step has been pairwise comparisons of the projects based on the effort and LOC criteria. The weights have been generated for the projects by using the FAHP methodology as discussed in Sect. 3.1.1 and using the fuzzy linguistic scale as described in Table 1. The weight vector generated by FAHP has been used as input to LSSVM model. The weight vector has been used to modify the kernel function as in Eq. (17) as discussed by Xing et al. (2009).

$$K(x_k, x_l) = \exp \frac{-||S(x_k - x_l)||^2}{\sigma^2} \tag{17}$$

In the equation, $||(x_k - x_l)||^2$ may be recognised as the squared Euclidean distance between the two feature vectors and S = diag[$\theta_1, \theta_2, ..., \theta_n$], where $\theta_1, \theta_2, ..., \theta_n$ are weight vectors generated by FAHP. However, as discussed by Guo et al. (2008) and Xing et al. (2009), the components of feature weights subject to the conditions, viz. $0 \leq \theta_k \leq 1$ where $k = 1, ..., n$ and $\sum_{k=1}^{n} \theta_k = 1$.

Further, LOC and effort values of the projects have been used to train LSSVM model. An optimal combination of parameters (γ,σ) has been considered where γ denotes the relative weights to allowed e_k errors during the training phase and σ is a kernel parameter. They have been tuned keeping in consideration fitness value of the problem. In this case, the fitness value to be optimised is MMRE (Eq. 19) and its value should be minimum for making more accurate predictions. For tuning of parameters, n-fold cross-validation has been used. After tuning of parameters, the trained model has been provided LOC as input and effort has been calculated. The effort value has been calculated for both the methods, viz. RBF kernel-based LSSVM (RBF-LSSVM) as given in Eq. (18) and FAHP modified RBF kernel-based LSSVM (FAHP-RBF-LSSVM) as in Eq. (17).

Algorithm 2: Algorithm used for development of hybrid model for modified kernel using FAHP

```
   Input: Comparison matrix, Fuzzy triangular number for finding
          the relative importance of criteria, Linguistic fuzzy scale,
          Measured effort, Tuning parameters ($\sigma$ and $\gamma$)
   Output: $E_{FAHP-RBF-SVM}$, $E_{RBF-SVM}$
   Result: Compute and outputs the effort using LSSVM and
           modified kernel LSSVM.
 1 Initialisation:
 2 Input the comparison matrix by the expert.
 3 Compute the pairwise comparison matrix.
 4 Compute the consistency of the matrix.
 5 if consistency ratio $\geq$ 0.1 then
 6     Go to line 2
 7 else
 8     Compute the synthetic extent values.
 9     Compute the degree of possibility for fuzzy numbers.
10     Compute the weight matrix for criteria.
11     Compute the normalised weighted decision matrix.
12 end
13 Input the measured effort data of known projects for a given
   environment.
14 Train the LSSVM models using optimal values of data $\sigma$ and $\gamma$.
   Here n-fold cross-validation has been used.
15 Compute RBF-SVM: $K(x_k, x_l) = \exp(-\frac{||x_k - x_l||^2}{\sigma^2})$ and
   FAHP-RBF-SVM: $K(x_k, x_l) = \exp \frac{-||S(x_k - x_l)||^2}{\sigma^2}$.
16 Compute the weighted LSSVM and generate the predicted effort.
17 Plot the functions.
18 return
```

Table 3 Description of attributes of COCOMO dataset

Category	Effort multiplier	Description
Product attributes	RELY	Required software reliability
	DATA	Data base size
	CPLX	Process complexity
Computer attributes	TIME	Time constraint for CPU
	STOR	Main memory constraint
	VIRT	Machine volatility
	TURN	Turnaround time
Personnel attributes	ACAP	Analysts capability
	PCAP	Programmers capability
	AEXP	Application experience
	VEXP	Virtual machine experience
	LEXP	Language experience
Project attributes	MODP	Modern programing practices
	TOOL	Use of software tools
	SCED	Schedule constraint

Table 4 Statistical analysis of COCOMO dataset

	LOC	Actual effort
Minimum	1.98	5.9
1st Quartile	8.65	40.5
Median	25.00	98.0
Mean	77.21	683.3
3rd Quartile	60.00	438.0
Maximum	150.00	11400.0

$$K(x_k, x_l) = \exp\left(-\frac{||x_k - x_l||^2}{\sigma^2}\right) \tag{18}$$

5 Empirical validation

For empirical validation of the algorithmic models, the datasets from the public repository and private vendor projects data have been used. The dataset of COCOMO, Kemerer and NASA is available in public domain (Menzies et al. 2012). The private vendor dataset has been taken from Srivastava et al. (2012).

5.1 Performance measures

The performance of different approaches has been evaluated using different established performance measures, mean magnitude of relative error (MMRE) and root-mean-square error (RMSE) as depicted in Eqs. (19) and (20), respectively. Here, the actual effort is taken from the dataset and predicted effort is the effort calculated using the proposed technique. These measures have been widely used by the research community.

$$\text{MMRE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{Actual effort} - \text{Predicted effort}}{\text{Actual effort}} \tag{19}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{Actual effort} - \text{Predicted effort})^2} \tag{20}$$

5.2 COCOMO dataset

COCOMO dataset is most commonly used dataset for evaluating the performance of proposed techniques. This dataset consists of 63 software projects, each of which is dependent upon 17 cost drivers (independent features). The effort value (dependent feature) is measured in person-months.

Table 3 presents the complete description of effort multipliers. Effort value (dependent feature) is measured in person-months and is dependent upon LOC and 15 effort multipliers. Each of the multipliers receives a rating on a six-point scale that ranges from “very low” to “extra high” (in importance or value).

The statistical analysis of COCOMO dataset is depicted in Table 4. From statistical analysis, it is evident that LOC

Table 5 Comparison matrix of COCOMO dataset with respect to effort criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	[1 1 1]	[1 2 3]	[3 4 5]	[3 4 5]	[5 6 7]	[5 6 7]	[8 9 9]	[2 3 4]	[3 4 5]	[3 4 5]
P2	0	[1 1 1]	[2 3 4]	[2 3 4]	[4 5 6]	[4 5 6]	[5 6 7]	[1 2 3]	[2 3 4]	[2 3 4]
P3	0	0	[1 1 1]	[1 1 1]	[2 3 4]	[2 3 4]	[4 5 6]	[1/4 1/3 1/2]	[1/3 1/2 1]	[1/3 1/2 1]
P4	0	0	0	[1 1 1]	[2 3 4]	[4 5 6]	[1/4 1/3 1/2]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]
P5	0	0	0	0	[1 1 1]	[1 1 1]	[2 3 4]	[1/7 1/6 1/5]	[1/4 1/3 1/2]	[1/4 1/3 1/2]
P6	0	0	0	0	0	[1 1 1]	[2 3 4]	[1/7 1/6 1/5]	[1/4 1/3 1/2]	[1/4 1/3 1/2]
P7	0	0	0	0	0	0	[1 1 1]	[1/8 1/7 1/6]	[1/7 1/6 1/5]	[1/6 1/5 1/4]
P8	0	0	0	0	0	0	0	[1 1 1]	[2 3 4]	[2 3 4]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[1 2 3]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 6 Comparison matrix of COCOMO dataset with respect to LOC criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	[1 1 1]	[1/3 1/2 1]	[1 1 1]	[1 2 3]	[4 5 6]	[5 6 7]	[5 6 7]	[4 5 6]	[4 5 6]	[4 5 6]
P2	0	[1 1 1]	[1 2 3]	[3 4 5]	[4 5 6]	[8 9 9]	[8 9 9]	[5 6 7]	[5 6 7]	[5 6 7]
P3	0	0	[1 1 1]	[1 2 3]	[4 5 6]	[5 6 7]	[5 6 7]	[4 5 6]	[4 5 6]	[4 5 6]
P4	0	0	0	[1 1 1]	[2 3 4]	[3 4 5]	[3 4 5]	[1 2 3]	[1 2 3]	[1 2 3]
P5	0	0	0	0	[1 1 1]	[1 2 3]	[1 2 3]	[1 1 1]	[1 1 1]	[1 1 1]
P6	0	0	0	0	0	[1 1 1]	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/3 1/2 1]
P7	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/3 1/3 1]	[1/3 1/2 1]
P8	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 1 1]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 7 Normalised weights

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Weight	0.191	0.289	0.191	0.096	0.045	0.026	0.026	0.046	0.046	0.046

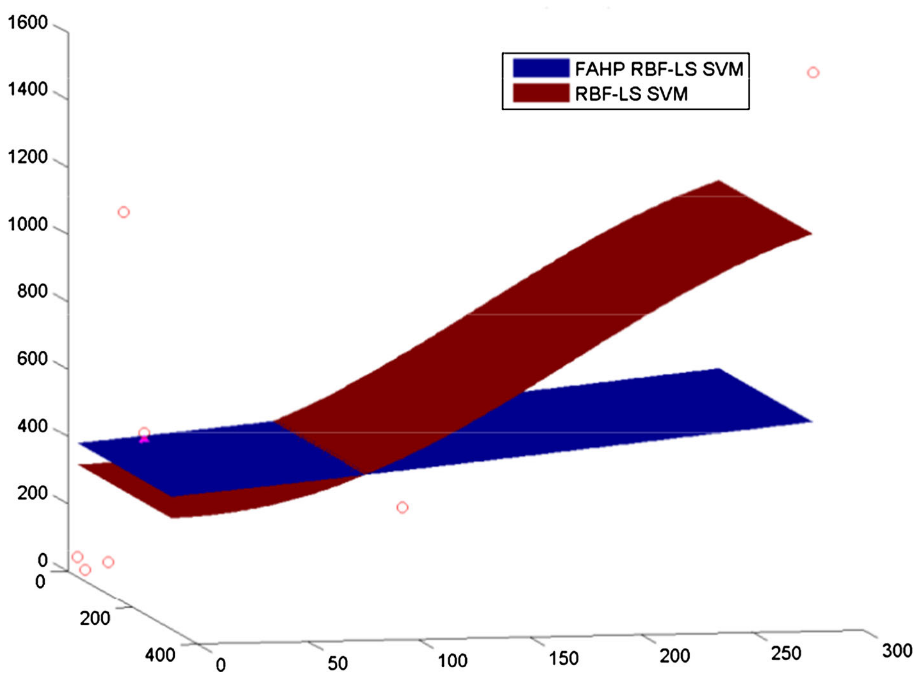


Fig. 3 Hyperplane created by using cross-validation and tuning parameters such that $\gamma = 1.4221$ and $\sigma^2 = 2.399$

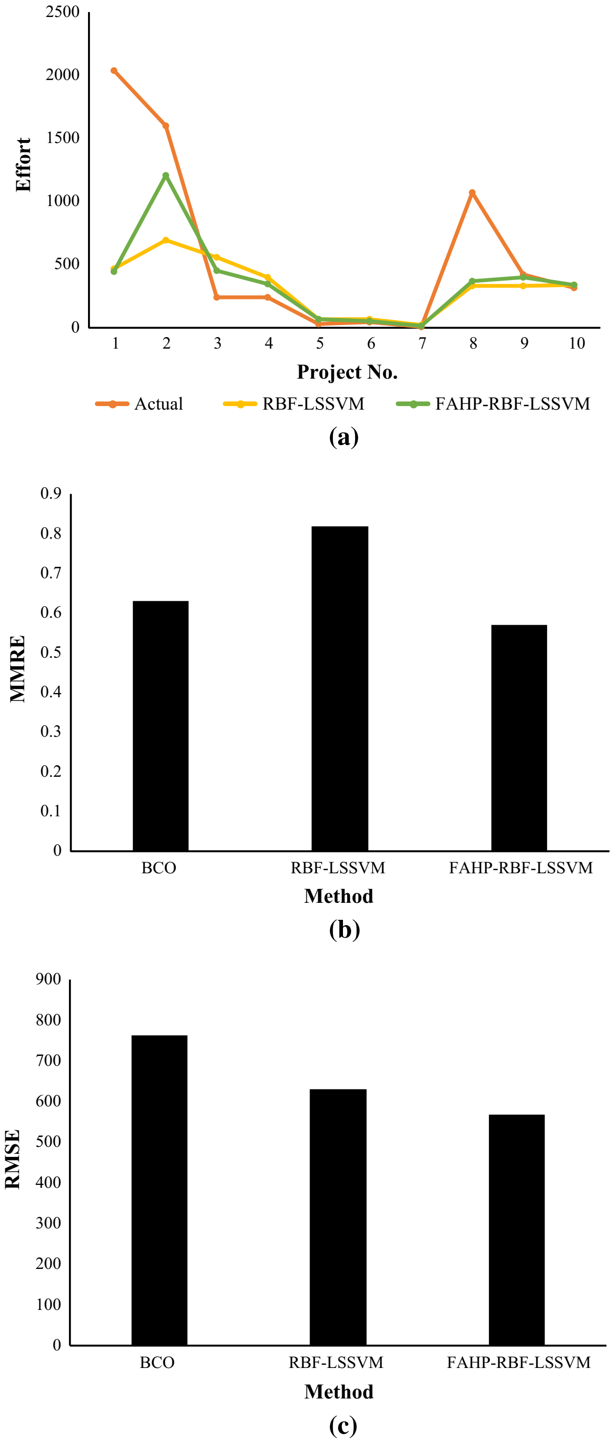
values of the software projects are dispersed over a range of values with minimum value as 1.98 and maximum value as 150. Similarly, effort values also span a wide range with minimum value as 5.9 and maximum value as 11400. Thus, these data present the development of diverse projects.

The experts have an opinion of giving equal importance to multiple criteria, viz. effort and LOC. Thus, based on the fuzzy scale presented in Table 1, the comparison matrices are constructed as given in Tables 5 and 6. Data of 10 projects from the COCOMO dataset consisting of 63 projects have been considered to represent the methodology used. It is considered that the matrix created would have been a complex one if all the projects have been considered for the pairwise comparisons. The comparison matrix has been created by taking effort and LOC as the criteria and projects as alternatives. The other parameters of the dataset have been taken as constant and have not been used for analysis. The projects have been relatively ranked based on the effort and LOC values. In Table 5, the first row represents the relative ranking of projects from P_1 to P_{10} with reference to project P_1. Similarly, relative ranks for all other projects have been generated by using comparative judgements by considering the effort as a criterion. In addition, the same methodology has been used to construct the comparison matrix as depicted in Table 6. The values of CR for the matrices based upon effort and LOC are 0.088 and 0.049, respectively, which are less than 0.1, and hence, these matrices are considered consistent for weight calculations. The ranks generated have been used for computing the weights by following the methodology of FAHP as discussed in Sect. 3.1.1. The normalised weights thus calculated are shown in Table 7.

The normalised weights thus obtained have been used to modify RBF kernel as given in Eq. (17). Further, the modified kernel has been used in LSSVM model to generate the effort values. As a numerical example, the value of effort has been computed for LOC value as 30. For this purpose,

Table 8 Effort comparison for RBF-LSSVM and FAHP-RBF-LSSVM using COCOMO dataset

Project no.	Actual effort	RBF-LSSVM	FAHP-RBF-LSSVM
1	2040	468.05	446.83
2	1600	697.51	1219.86
3	243	562.31	454.20
4	240	400.54	346.07
5	33	70.45	65.34
6	43	67.56	56.67
7	8	25.35	18.23
8	1075	330.58	371.68
9	423	329.83	396.74
10	321	341.76	340.27



(a)

(b)

(c)

Fig. 4 Empirical validation of hybrid model using COCOMO dataset. **a** Effort comparison. **b** MMRE comparison. **c** RMSE comparison

Table 9 MMRE and RMSE comparison of BCO, RBF-LSSVM and FAHP-RBF-LSSVM using COCOMO dataset

S. no.	Method	MMRE
MMRE		
1	BCO	0.63
2	RBF-LSSVM	0.82
3	FAHP-RBF-LSSVM	0.57
RMSE		
1	BCO	763.07
2	RBF-LSSVM	630.78
3	FAHP-RBF-LSSVM	569.43

Table 10 Statistical analysis of NASA dataset

	DL	ME	Effort
Minimum	2.1	19	5
1st Quartile	8.275	26	9.325
Median	17.15	28.5	26.2
Mean	33.589	27.78	49.472
3rd Quartile	52.5	31	94.7
Maximum	100.8	35	138.3

the values of LOC and effort of remaining projects other have been provided as training data. The normalised weights obtained earlier have been given as the input to modify the kernel function as represented in Eq. (17). Then, LOC value of 30 has been provided as input and the values of γ and σ have been tuned to minimise MMRE. The values of γ and σ^2 have been tuned to 1.4221 and 2.399, respectively. The effort value computed for LOC value of 30 is 329.83 and 396.74 for methods RBF-LSSVM and FAHP-RBF-LSSVM, respectively. This has been presented in the form of a hyperplane as illustrated in Fig. 3. The effort values for other projects have been computed in the same manner. The effort values obtained by modified RBF kernel and unmodified RBF kernel are presented in Table 8 and Fig. 4a.

Table 9 depicts the MMRE and RMSE comparison values for RBF-LSSVM and FAHP-RBF-LSSVM with BCO. MMRE and RMSE values for FAHP-RBF-LSSVM are 0.57 and 569.43, respectively, proving the outperformance of FAHP-RBF-LSSVM over RBF-LSSVM and BCO. MMRE and RMSE comparison is also depicted in Fig. 4b, c.

5.3 NASA dataset

NASA dataset is composed of 4 attributes out of which 2 are independent attributes, viz. methodology (ME) and developed lines (DL). DL depicts program size including both new source code and reused code from other projects. It is presented in KLOC (thousands of lines of codes). The ME attribute is related to the methodology used in the development of each software project. The dataset consists of 1 dependent attribute; namely, effort is given in man-months. The statistical analysis of NASA dataset is presented in Table 10. DL attribute has minimum value as 2.1 and max-

Table 11 Comparison matrix of NASA dataset with respect to effort criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
P1	[1 1 1]	[1 2 3]	[2 3 4]	[1 2 3]	[4 5 6]	[1 2 3]	[6 7 8]	[7 8 9]	[4 5 6]	[7 8 9]	[7 8 9]	[7 8 9]	[7 8 9]	[7 8 9]	[1 2 3]	[6 7 8]	[4 5 6]	[1/3 1/2 1]
P2	0	[1 1 1]	[1 2 3]	[1 1 1]	[2 3 4]	[1 1 1]	[4 5 6]	[4 5 6]	[3 4 5]	[5 6 7]	[5 6 7]	[5 6 7]	[5 6 7]	[5 6 7]	[1 1 1]	[4 5 6]	[3 4 5]	[1/4 1/3 1/2]
P3	0	0	[1 1 1]	[1/3 1/2 1]	[3 4 5]	[1/3 1/2 1]	[3 4 5]	[6 7 8]	[2 3 4]	[4 5 6]	[4 5 6]	[4 5 6]	[4 5 6]	[4 5 6]	[1/3 1/2 1]	[3 4 5]	[2 3 4]	[1/6 1/5 1/4]
P4	0	0	0	[1 1 1]	[2 3 4]	[1 1 1]	[4 5 6]	[4 5 6]	[3 4 5]	[5 6 7]	[5 6 7]	[5 6 7]	[5 6 7]	[5 6 7]	[1 1 1]	[4 5 6]	[3 4 5]	[1/4 1/3 1/2]
P5	0	0	0	0	[1 1 1]	[1/4 1/3 1/2]	[2 3 4]	[2 3 4]	[1 2 3]	[2 3 4]	[2 3 4]	[2 3 4]	[2 3 4]	[2 3 4]	[1/4 1/3 1/2]	[2 3 4]	[1 2 3]	[1/8 1/7 1/6]
P6	0	0	0	0	0	[1 1 1]	[4 5 6]	[4 5 6]	[3 4 5]	[5 6 7]	[5 6 7]	[5 6 7]	[5 6 7]	[5 6 7]	[1 1 1]	[4 5 6]	[3 4 5]	[1/4 1/3 1/2]
P7	0	0	0	0	0	0	[1 1 1]	[1 2 3]	[1/3 1/2 1]	[2 3 4]	[2 3 4]	[2 3 4]	[2 3 4]	[2 3 4]	[1/6 1/5 1/4]	[1 1 1]	[1/3 1/2 1]	[1/9 1/9 1/8]
P8	0	0	0	0	0	0	0	[1 1 1]	[1/4 1/3 1/2]	[1 1 1]	[1 1 1]	[1 1 1]	[1 1 1]	[1 1 1]	[1/9 1/8 1/7]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/9 1/9 1/8]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[3 4 5]	[2 3 4]	[2 3 4]	[2 3 4]	[2 3 4]	[1/5 1/4 1/3]	[1 2 3]	[1 1 1]	[1/7 1/6 1/5]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 1 1]	[1 1 1]	[1 1 1]	[1/9 1/8 1/7]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/9 1/9 1/8]
P11	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 1 1]	[1 1 1]	[1/9 1/8 1/7]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/9 1/9 1/8]
P12	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 1 1]	[1/9 1/8 1/7]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/9 1/9 1/8]
P13	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1/9 1/8 1/7]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/9 1/9 1/8]
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/9 1/9 1/8]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/9 1/9 1/8]
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[6 7 8]	[4 5 6]	[1/4 1/3 1/2]
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/9 1/8 1/7]
P17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/5 1/4 1/3]
P18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 12 Comparison matrix of NASA dataset with respect to LOC criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
P1	[1 1 1]	[2 3 4]	[2 3 4]	[2 3 4]	[4 5 6]	[2 3 4]	[5 6 7]	[5 6 7]	[4 5 6]	[6 7 8]	[6 7 8]	[5 6 7]	[6 7 8]	[6 7 8]	[1 2 3]	[5 6 7]	[5 6 7]	[1/3 1/2 1]
P2	0	[1 1 1]	[1 1 1]	[1/3 1/2 1]	[1 2 3]	[1/4 1/3 1/2]	[2 3 4]	[2 3 4]	[1 2 3]	[3 4 5]	[3 4 5]	[2 3 4]	[3 4 5]	[3 4 5]	[1/3 1/2 1]	[2 3 4]	[2 3 4]	[1/4 1/3 1/2]
P3	0	0	[1 1 1]	[1/3 1/2 1]	[1 2 3]	[1/4 1/3 1/2]	[2 3 4]	[2 3 4]	[1 2 3]	[3 4 5]	[3 4 5]	[2 3 4]	[3 4 5]	[3 4 5]	[1/3 1/2 1]	[2 3 4]	[2 3 4]	[1/4 1/3 1/2]
P4	0	0	0	[1 1 1]	[1 2 3]	[1/3 1/2 1]	[2 3 4]	[3 4 5]	[2 3 4]	[4 5 6]	[4 5 6]	[3 4 5]	[3 4 5]	[3 4 5]	[1/3 1/2 1]	[3 4 5]	[3 4 5]	[1/3 1/2 1]
P5	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[2 3 4]	[2 3 4]	[1 2 3]	[3 4 5]	[3 4 5]	[2 3 4]	[3 4 5]	[3 4 5]	[1/4 1/3 1/2]	[2 3 4]	[2 3 4]	[1/5 1/4 1/3]
P6	0	0	0	0	0	[1 1 1]	[4 5 6]	[4 5 6]	[2 3 4]	[4 5 6]	[4 5 6]	[3 4 5]	[4 5 6]	[4 5 6]	[1/3 1/2 1]	[3 4 5]	[3 4 5]	[1/3 1/2 1]
P7	0	0	0	0	0	0	[1 1 1]	[1 2 3]	[1/3 1/2 1]	[2 3 4]	[2 3 4]	[1 1 1]	[2 3 4]	[2 3 4]	[1/6 1/5 1/4]	[1 1 1]	[1 1 1]	[1/7 1/6 1/5]
P8	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[2 3 4]	[2 3 4]	[1 1 1]	[2 3 4]	[2 3 4]	[1/6 1/5 1/4]	[1 1 1]	[1 1 1]	[1/7 1/6 1/5]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[2 3 4]	[2 3 4]	[1 2 3]	[2 3 4]	[2 3 4]	[1/4 1/3 1/2]	[2 3 4]	[2 3 4]	[1/6 1/5 1/4]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1/3 1/2 1]	[1 1 1]	[1 1 1]	[1/7 1/6 1/5]	[1/3 1/2 1]	[1/3 1/2 1]	[1/8 1/7 1/6]
P11	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1 1 1]	[1 1 1]	[1/7 1/6 1/5]	[1/3 1/2 1]	[1/3 1/2 1]	[1/8 1/7 1/6]
P12	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[2 3 4]	[1 1 1]	[1/7 1/6 1/5]	[1 1 1]	[1/4 1/3 1/2]	[1/8 1/7 1/6]
P13	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1/8 1/7 1/6]	[1/3 1/2 1]	[1/3 1/2 1]	[1/7 1/6 1/5]
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/9 1/9 1/8]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/9 1/9 1/8]
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/7 1/6 1/5]
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[2 3 4]	[2 3 4]	[1/3 1/2 1]
P17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/6 1/5 1/4]
P18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 13 Effort comparison of RBF-LSSVM and FAHP-RBF-LSSVM using NASA dataset

Project no.	Actual effort	RBF-LSSVM	FAHP-RBF-LSSVM
1	115.8	71.09	102.34
2	96	75.9	89.93
3	79	75.16	80.47
4	90.8	80.29	84.98
5	39.6	75.6	43.32
6	98.4	65.79	93.91
7	18.9	16.35	16.44
8	10.3	15.86	14.7
9	28.5	17.33	22.08
10	7	13.18	9.44
11	9	13.77	10.05
12	7.3	15.53	12.91
13	5	9.7	3.8
14	8.4	15.54	10.89
15	98.7	70.91	101.3
16	15.6	13.24	15.26
17	23.9	48.8	32.77
18	138.3	157.43	125.67

Table 14 MMRE and RMSE comparison of BCO, RBF-LSSVM and FAHP-RBF-LSSVM using NASA dataset

S. no.	Method	MMRE
MMRE		
1	BCO	0.21
2	RBF-LSSVM	0.5
3	FAHP-RBF-LSSVM	0.19
RMSE		
1	BCO	9.9
2	RBF-LSSVM	19.74
3	FAHP-RBF-LSSVM	5.99

imum value as 100.8, and effort attribute spans the range with minimum value as 5 and maximum value as 138.3. This dataset is less diverse as compared to COCOMO dataset.

The projects of NASA dataset are compared on the basis of effort and LOC values given in the dataset based on the linguistic scale as given in Table 1. The comparison matrix based on effort values is depicted in Table 11. CR value of this matrix has been 0.097 which is less than 0.1; thus, this matrix is considered as a consistent matrix. The comparison matrix based on LOC values is depicted in Table 12. The value of CR for this matrix has been 0.097 justifying the consistency of the matrix.

The effort values have been computed using the RBF kernel function and modified RBF kernel function by weight vector generated using FAHP. The comparison of values obtained using RBF-LSSVM and FAHP-RBF-LSSVM is presented in Table 13. Table 14 depicts the MMRE and RMSE values, respectively, for NASA dataset. MMRE and RMSE values for FAHP-RBF-LSSVM are 0.19 and 5.99,

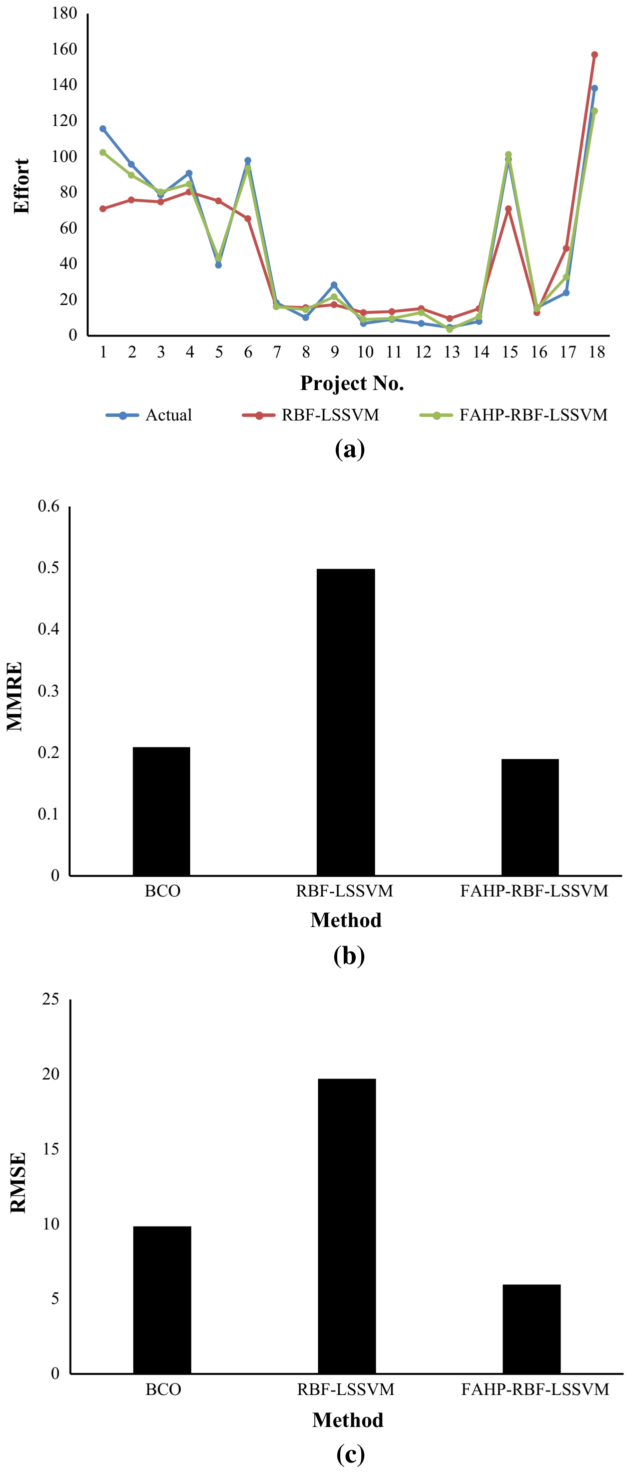


Fig. 5 Empirical validation of hybrid model using NASA dataset. **a** Effort comparison. **b** MMRE comparison. **c** RMSE comparison

respectively. These are comparatively less than RBF-LSSVM and BCO technique. Thus, it clearly depicts that weighted RBF kernel using FAHP performs better as compared to other methods. Figure 5a–c presents the comparison of effort, MMRE and RMSE in pictorial form.

5.4 Kemerer dataset

Kemerer dataset consists of 15 software projects characterised by 6 independent and 1 dependent attribute. Table 15 presents the description of the attributes. The independent attributes include 2 categorical and 4 numerical features. The categorical attributes are "Language" and "Hardware", and numerical attributes are "Duration", "KSLOC", "AdjFP" and "RawFP". Effort (dependent attribute) is measured by 'man-months'.

The statistical analysis of Kemerer dataset is depicted in Table 16. Numerical attributes have been analysed statistically, and KSLOC varies from 39 to 450. The effort value spans the range from 23.2 to 1107.31. This dataset is of less diverse projects that are compared to the previously discussed datasets.

The comparison matrices have been obtained from effort and LOC values of 15 projects of Kemerer dataset. The matrices based upon effort and LOC values, respectively, are presented in Tables 17 and 18. The values of CR for the matrices based upon effort and loc are 0.096 and 0.067, respectively. Since the values are less than 0.1, thus matrices are considered as consistent matrices. The effort values computed using FAHP-RBF-LSSVM and RBF-LSSVM are presented in Table 19.

Table 15 Description of attributes of Kemerer dataset

Attribute	Description
KSLOC	Kilo lines of code
AdjFP	Adjusted function points
RAWFP	Unadjusted function points
Duration	Duration of project
Language	Programming language
Hardware	Hardware resources

Table 16 Statistical analysis of Kemerer dataset

	Duration	KSLOC	AdjFP	RawFP	Effort
Minimum	5	39	99.9	97	23.2
1st Quartile	9	55.1	599.1	581	83.25
Median	14	164.8	993	976	130.3
Mean	14.27	186.6	999.1	993.9	219.25
3rd Quartile	17.5	253.9	1342.5	1464.5	252.8
Maximum	31	450	2306.8	2284	1107.31

The comparison of MMRE and RMSE values for RBF-LSSVM and FAHP-RBF-LSSVM is presented in Table 20 and Fig. 6a–c. The results clearly reveal that incorporation of weight vector generated by FAHP -RBF-LSSVM has resulted in better effort estimates. MMRE and RMSE values obtained for FAHP-RBF-LSSVM are 0.31 and 149.5, respectively, showing the dominance over other methods.

5.5 Interactive voice response (IVR) dataset

The data in IVR dataset have been collected of IVR application though survey from a software industry where IVR application is developed with questionnaires directly to the company's project managers and senior software development professionals (Srivastava et al. 2012). It consists of 4 attributes, viz. project no., KLOC, actual effort and actual time.

The statistical analysis of IVR dataset is presented in Table 21. From Table 21, it is evident that the values of the attributes are not spread over a wide range. This reveals that projects are of similar category.

The comparison matrix of IVR dataset has been obtained by comparing the effort values of 10 projects from the dataset as comparison matrix of 48 projects would have been a complex matrix. The alternatives have been relatively ranked based on multiple criteria, viz. LOC and effort value. Based on the dataset, the expert has given equal importance to LOC and effort for their relative ranking. The comparison matrices thus created are depicted in Tables 22 and 23. It is found that the created comparison matrices based upon effort and LOC have the CR value as 0.039 and 0.045, respectively, which is less than 0.1. Hence, the comparison matrices are considered consistent for further calculations. Effort values computed using RBF-LSSVM and FAHP-RBF-LSSVM are shown in Table 25.

Figure 7a–c depicts the comparison of effort, MMRE and RMSE values of RBF-LSSVM, FAHP-RBF-LSSVM and BCO. Table 24 reveals the values of MMRE and RMSE as 0.07 and 5.23 for FAHP-RBF-LSSVM which are less than the values for RBF-LSSVM and BCO showing the dominance of FAHP-RBF-LSSVM.

6 Conclusion and future scope

It has been identified that SEE acts as a base point for many project management activities including planning, budgeting and scheduling. Thus, it is crucial to obtain almost accurate estimates. The researchers have proposed numerous estimation methods since the inception of SE as a research area. Further, it has been witnessed that SEE process depends on multiple intrinsic and extrinsic factors. Despite extensive research on it, the community is unable to develop and accept

Table 17 Comparison matrix of Kemerer dataset with respect to effort criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
P1	[1 1 1]	[2 3 4]	[1/4 13 1/2]	[2 3 4]	[1/3 1/2 1]	[2 3 4]	[4 5 6]	[1 2 3]	[1 2 3]	[2 3 4]	[1 1 1]	[1 1 1]	[1 2 3]	[1 1 1]	[2 3 4]
P2	0	[1 1 1]	[1/5 1/4 1/3]	[1 1 1]	[1/3 1/2 1]	[1 1 1]	[2 3 4]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1 2 3]
P3	0	0	[1 1 1]	[3 4 5]	[1 2 3]	[3 4 5]	[6 7 8]	[4 5 6]	[4 5 6]	[3 4 5]	[2 3 4]	[2 3 4]	[4 5 6]	[2 3 4]	[3 4 5]
P4	0	0	0	[1 1 1]	[1/3 1/2 1]	[1 1 1]	[2 3 4]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]	[1/4 1/3 1/2]	[1/3 1/2 1]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1 2 3]
P5	0	0	0	0	[1 1 1]	[1 2 3]	[3 4 5]	[1 2 3]	[1 2 3]	[1 2 3]	[1 1 1]	[1 1 1]	[1 2 3]	[1 1 1]	[3 4 5]
P6	0	0	0	0	0	[1 1 1]	[2 3 4]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/3 1/2 1]	[1/4 1/3 1/2]	[2 3 4]
P7	0	0	0	0	0	0	[1 1 1]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/3 1/2 1]	[1/6 1/5 1/4]	[1/6 1/5 1/4]	[1/7 1/6 1/5]	[1/6 1/5 1/4]	[1/3 1/2 1]
P8	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 2 3]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]	[1/3 1/2 1]	[2 3 4]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[1 2 3]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]	[1/3 1/2 1]	[1 2 3]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1 1 1]
P11	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 2 3]	[1 1 1]	[4 5 6]
P12	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 2 3]	[1 1 1]	[4 5 6]
P13	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1 2 3]
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[2 3 4]
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 18 Comparison matrix of Kemerer dataset with respect to LOC criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
P1	[1 1 1]	[3 4 5]	[1/4 1/3 1/2]	[1 1 1]	[1/4 1/3 1/2]	[3 4 5]	[3 4 5]	[1 1 1]	[1 1 1]	[3 4 5]	[1 1 1]	[2 3 4]	[1 2 3]	[1 2 3]	[3 4 5]
P2	0	[1 1 1]	[1/7 1/6 1/5]	[1/5 1/4 1/3]	[1/7 1/6 1/5]	[1 1 1]	[1 1 1]	[1/5 1/4 1/3]	[1/7 1/6 1/5]	[1 1 1]	[1/5 1/4 1/3]	[1/5 1/4 1/3]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1 1 1]
P3	0	0	[1 1 1]	[2 3 4]	[1 1 1]	[6 7 8]	[6 7 8]	[2 3 4]	[2 3 4]	[5 6 7]	[2 3 4]	[4 5 6]	[3 4 5]	[3 4 5]	[5 6 7]
P4	0	0	0	[1 1 1]	[1/4 1/3 1/2]	[3 4 5]	[3 4 5]	[1 1 1]	[1 1 1]	[3 4 5]	[1 1 1]	[2 3 4]	[1 2 3]	[1 2 3]	[3 4 5]
P5	0	0	0	0	[1 1 1]	[6 7 8]	[6 7 8]	[2 3 4]	[2 3 4]	[5 6 7]	[2 3 4]	[4 5 6]	[3 4 5]	[3 4 5]	[5 6 7]
P6	0	0	0	0	0	[1 1 1]	[1 1 1]	[1/5 1/4 1/3]	[1/7 1/6 1/5]	[1 1 1]	[1/5 1/4 1/3]	[1/5 1/4 1/3]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1 1 1]
P7	0	0	0	0	0	0	[1 1 1]	[1/5 1/4 1/3]	[1/7 1/6 1/5]	[1 1 1]	[1/5 1/4 1/3]	[1/5 1/4 1/3]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1 1 1]
P8	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[4 5 6]	[1 1 1]	[3 4 5]	[1 2 3]	[1 2 3]	[4 5 6]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[3 4 5]	[1 1 1]	[3 4 5]	[2 3 4]	[2 3 4]	[4 5 6]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/5 1/4 1/3]	[1/5 1/4 1/3]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1 1 1]
P11	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[2 3 4]	[1 2 3]	[1 2 3]	[4 5 6]
P12	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1 2 3]
P13	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1 2 3]
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]	[1 2 3]
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 19 Effort comparison for RBF-LSSVM and FAHP-RBF-LSSVM using Kemerer dataset

Project no.	Actual effort	RBF-LSSVM	FAHP-RBF-LSSVM
1	287	270.28	279.28
2	82.5	390.14	285.84
3	1107.31	376.62	631.52
4	86.9	383.23	312
5	336.3	563.44	338.11
6	84	65.61	82.99
7	23.2	85.57	98.84
8	130.3	101.33	115.46
9	116	101.07	112.56
10	72	83.63	98.32
11	258.7	185.1	179.89
12	230.7	175.37	181.5
13	157	202.87	180.07
14	246.9	213.43	222.83
15	69.9	92.56	89.63

Table 20 MMRE and RMSE comparison of BCO, RBF-LSSVM and FAHP-RBF-LSSVM using Kemerer dataset

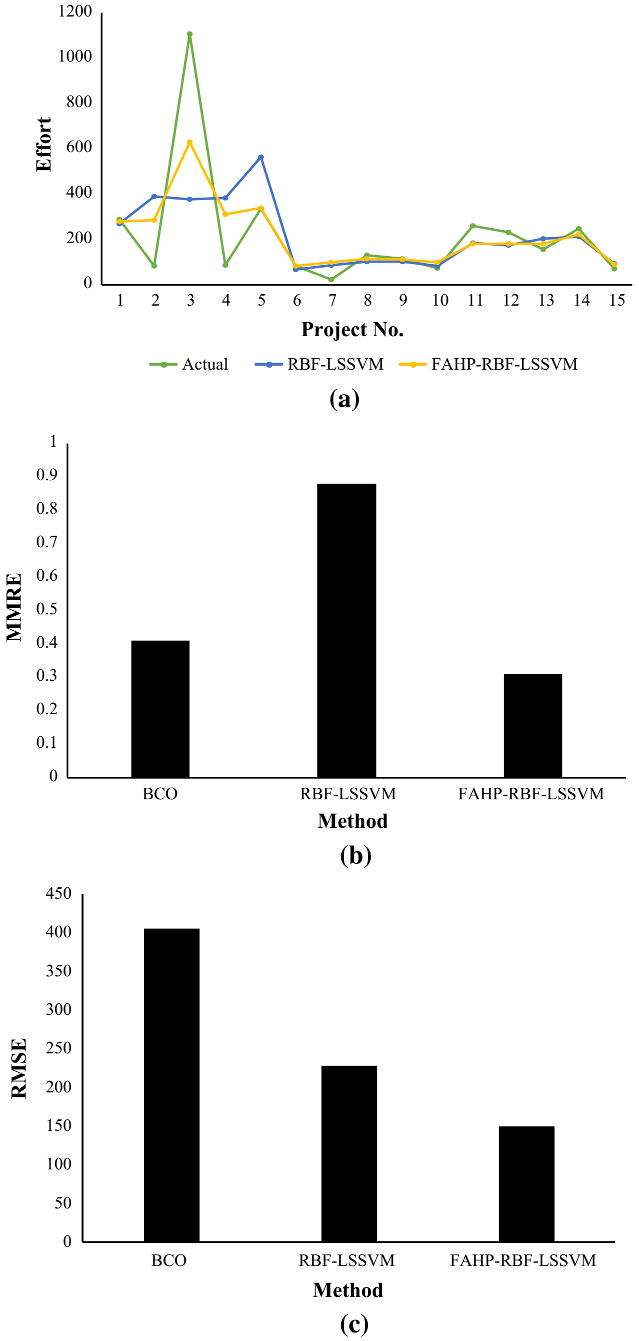
S.No.	Method	MMRE
MMRE		
1	BCO	0.41
2	RBF-LSSVM	0.88
3	FAHP-RBF-LSSVM	0.31
RMSE		
1	BCO	405.98
2	RBF-LSSVM	228.29
3	FAHP-RBF-LSSVM	149.5

Table 21 Statistical analysis of IVR dataset

	KLOC	Effort	Time
Minimum	2.500	9.40	4.000
1st Quartile	5.305	23.07	6.000
Median	6.65	30.92	6.500
Mean	7.161	34.13	6.565
3rd Quartile	8.5	41.15	7.125
Maximum	18	97.19	9.500

a single model that can be applied in diverse environments and which can handle multiple environmental factors.

Therefore, MCDM has been utilised for the process of SEE. FAHP, an extension of AHP, has been proposed in the current research for the purpose. It has been identified that FAHP can handle subjectivity and uncertainty using fuzzy numbers and CR.



(a)

(b)

(c)

Fig. 6 Empirical validation of hybrid model using Kemerer dataset. **a** Effort comparison. **b** MMRE comparison. **c** RMSE comparison

Further, to provide a robust method, a hybrid model has been developed to combine MCDM (for handling uncertainty) and ML algorithm (for handling imprecision) approach to predict the effort more accurately. The hybrid model has amalgamated MCDM approach (FAHP) and ML approach (LSSVM). FAHP has been used to generate feature weights. The weights have been calculated using effort and LOC as the criteria. The projects have been considered as alternatives.

Table 22 Comparison matrix of IVR dataset with respect to effort criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	[1 1 1]	[2 3 4]	[1 2 3]	[3 4 5]	[6 7 8]	[2 3 4]	[1 2 3]	[2 3 4]	[1 2 3]	[2 3 4]
P2	0	[1 1 1]	[1/3 1/2 1]	[1 1 1]	[1 2 3]	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]
P3	0	0	[1 1 1]	[1 2 3]	[2 3 4]	[1 2 3]	[1 1 1]	[1 2 3]	[1 1 1]	[1 2 3]
P4	0	0	0	[1 1 1]	[1 2 3]	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]
P5	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/3 1/2 1]
P6	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]
P7	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 1 1]	[1 2 3]
P8	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 1 1]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[1 2 3]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 23 Comparison matrix of IVR dataset with respect to LOC criterion

Project no.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	[1 1 1]	[2 3 4]	[1 2 3]	[2 3 4]	[3 4 5]	[2 3 4]	[1 2 3]	[1 2 3]	[1 2 3]	[2 3 4]
P2	0	[1 1 1]	[1/3 1/2 1]	[1 2 3]	[2 3 4]	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]
P3	0	0	[1 1 1]	[2 3 4]	[3 4 5]	[1 2 3]	[1 1 1]	[1 1 1]	[1 1 1]	[1 2 3]
P4	0	0	0	[1 1 1]	[1 2 3]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/3 1/2 1]
P5	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/4 1/3 1/2]	[1/3 1/2 1]
P6	0	0	0	0	0	[1 1 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1/3 1/2 1]	[1 1 1]
P7	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 1 1]	[1 2 3]
P8	0	0	0	0	0	0	0	[1 1 1]	[1 1 1]	[1 2 3]
P9	0	0	0	0	0	0	0	0	[1 1 1]	[1 2 3]
P10	0	0	0	0	0	0	0	0	0	[1 1 1]

Table 24 MMRE and RMSE comparison for BCO, RBF-LSSVM and FAHP-RBF-LSSVM using IVR dataset

S. No.	Method	MMRE
MMRE		
1	BCO	0.11
2	RBF-LSSVM	0.17
3	FAHP-RBF-LSSVM	0.07
RMSE		
1	BCO	10.03
2	RBF-LSSVM	8.15
3	FAHP-RBF-LSSVM	5.23

The ranks generated by FAHP have been utilised to modify the kernel. A weighted kernel (W_k) has been used to assimilate generated weights into the LSSVM. The performance of the proposed model has been compared with BCO. The combination of feature ranking by FAHP and the weighted kernel has resulted in more accurate effort estimates.

Table 25 Effort comparison for RBF-LSSVM and FAHP-RBF-LSSVM using IVR dataset

Project no.	Actual effort	RBF-LSSVM	FAHP-RBF-LSSVM
1	86.1	67.32	79.34
2	24.02	24.12	23.68
3	36.05	24.25	29.09
4	20.74	22.09	21.05
5	12.85	18.76	16.2
6	23.3	23.42	23.38
7	31.72	25.13	29.32
8	29.59	24.06	27.6
9	33.88	26.06	31.68
10	24.34	23.83	24.26

Future work may focus on the development of hybrid approach in which weights generated from other MCDM techniques can be amalgamated into ML techniques. Also, the prediction accuracy can be analysed by applying

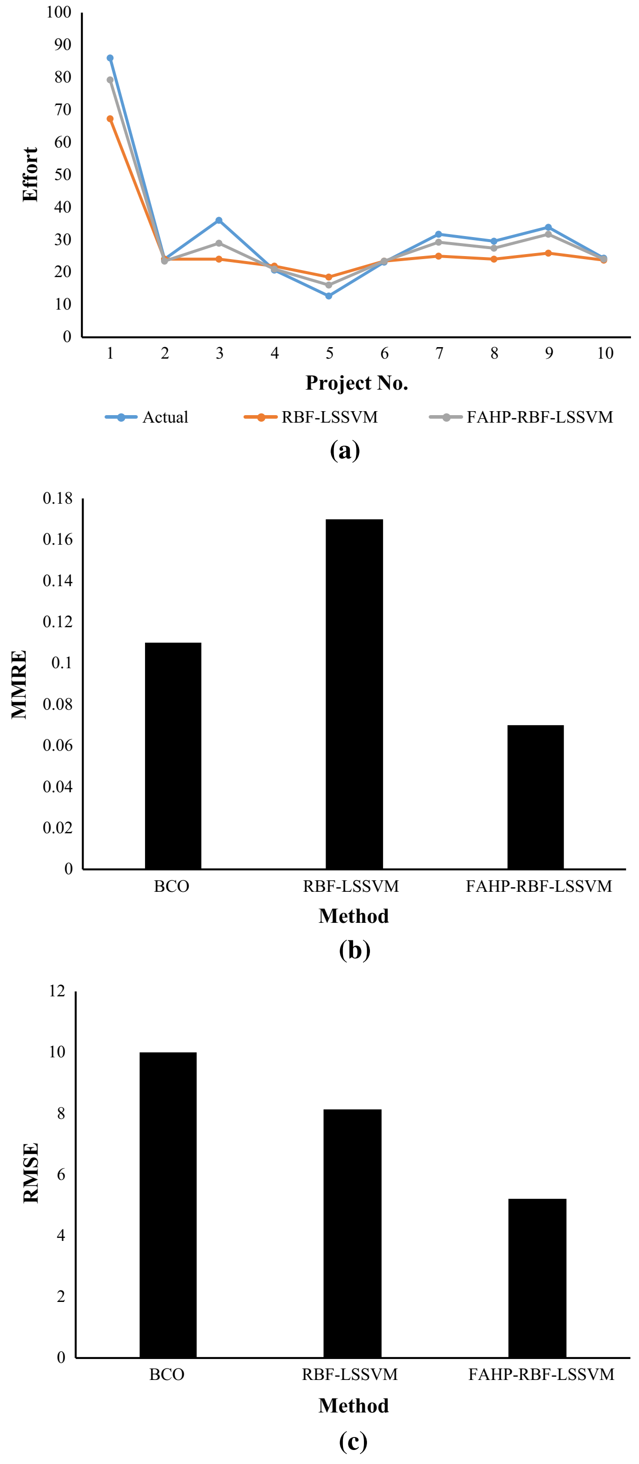


Fig. 7 Empirical validation of hybrid model using IVR dataset. **a** Effort comparison. **b** MMRE comparison. **c** RMSE comparison

other kernel-based functions modified by incorporating the weights generated by MCDM approaches.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Attarzadeh I, Ow SH (2009) Software development effort estimation based on a new fuzzy logic model. Int J Comput Theory Eng 1(4):473–476

Azzeh M, Nassif AB (2016) A hybrid model for estimating software project effort from use case points. Appl Soft Comput 49:981–989

Belton V, Stewart T (2002) The multiple criteria problem. Multiple criteria decision analysis: an integrated approach. Springer, Cham, pp 13–33

Benala TR, Bandarupalli R (2016) Least square support vector machine in analogy-based software development effort estimation. In: International conference on recent advances and innovations in engineering

Braga PL, Oliveira ALI, Ribeiro GHT, Meira SRL (2007) Bagging predictors for estimation of software project effort. In: International joint conference on neural networks. IEEE, Florida, USA, pp 1595–1600

Buckley J (1985) Fuzzy hierarchical analysis. Fuzzy Sets Syst 17(3):233–247

Chalotra S, Sehra S, Brar Y, Kaur N (2015a) Tuning of cocomo model parameters by using bee colony optimization. Indian J Sci Technol 8(14):1–5

Chalotra S, Sehra SK, Sehra SS (2015b) An analytical review of nature inspired optimization algorithms. Int J Sci Technol Eng 2(2):123–126

Chang D-Y (1992) Extent analysis and synthetic decision, optimization techniques and applications. World Sci 1(1):352–355

Chang D-Y (1996) Applications of the extent analysis method on fuzzy AHP. Eur J Oper Res 95(3):649–655

Corazza A, Martino SD, Ferrucci F, Gravino C, Mendes E (2011) Investigating the use of support vector regression for web effort estimation. Empir Softw Eng 16(2):211–243

Csutora R, Buckley JJ (2001) Fuzzy hierarchical analysis: the lambda-max method. Fuzzy Sets Syst 120(2):181–195

Dasheng X, Shenglan H (2012) Estimation of project costs based on fuzzy neural network. In: World congress on information and communication technologies. IEEE, Trivandrum, India, pp 1177–1181

Dave VS, Dutta K (2011) Comparison of regression model, feed-forward neural network and radial basis neural network for software development effort estimation. ACM SIGSOFT Softw Eng Notes 36(5):1–5

Ferrucci F, Gravino C, Sarro F (2011) How multi-objective genetic programming is effective for software development effort estimation? Search based software engineering. Springer, New York, pp 274–275

Furulund KM, Molokken-Ostvold K (2007) Increasing software effort estimation accuracy using experience data, estimation models and checklists. In: Seventh international conference on quality software. IEEE, Portland, USA, pp 342–347

Gestel Suykens JA, Baesens B, Viaene S, Vanthienen J, Dedene G, de Moor B, Vandewalle J (2004) Benchmarking least squares support vector machine classifiers. Mach Learn 54:5–32

Gharehchopogh FS, Rezaii R, Arasteh B (2015) A new approach by using Tabu search and genetic algorithms in software cost estimation. In: 9th International conference on application of information and communication technologies. IEEE, Rostov-on-Don, Russia, pp 113–117

Glass RL, Vessey I, Ramesh V (2002) Research in software engineering: an analysis of the literature. Inf Softw Technol 44(8):491–506

Guo B, Gunn SR, Damper RI, Nelson JDB (2008) Customizing kernel functions for SVM-based hyperspectral image classification. IEEE Trans Image Process 17(4):622–629

Hidmi O, Sakar BE (2017) Software development effort estimation using ensemble machine learning. Int J Comput Commun Instrum Eng 4(1):143–147

Idri A, Abran A, Mbarki S (2006) An experiment on the design of radial basis function neural networks for software cost estimation. In: International conference on information & communication technologies: from theory to applications, vol 2. IEEE, Damascus, Syria, pp 1612–1617

Idri A, Khoshgoftaar TM, Abran A (2002) Can neural networks be easily interpreted in software cost estimation? In: IEEE international conference on fuzzy systems FUZZ-IEEE'02, vol 2. IEEE, Honolulu, Hawaii, pp 1162–1167

Idri A, Zahi A, Mendes E, Zakrani A (2007) Software cost estimation models using radial basis function neural networks. In: Cuadrado-Gallego JJ, Braungarten RB, Dumke RR, Arban A (eds) Software process and product measurements, vol 4895. Lecture notes in computer science. Springer, Berlin, pp 21–31

Jiang Z, Naudé P (2007) An examination of the factors influencing software development effort. Int J Comput Inf Syst Sci Eng 1(4):182–191

Jørgensen M, Shepperd M (2007) A systematic review of software development cost estimation studies. IEEE Trans Softw Eng 33(1):33–53

Jørgensen M, Boehm B, Rifkin S (2009) Software development effort estimation: Formal models or expert judgment? IEEE Softw 26(2):14–19

Kahraman C, Cebeci U, Ulukan Z (2003) Multi-criteria supplier selection using fuzzy AHP. Logist Inf Manag 16(6):382–394

Kocaguneli E, Menzies T, Mendes E (2015) Transfer learning in effort estimation. Emp Softw Eng 20(3):813–843

Kuswandari R (2004) Assessment of different methods for measuring the sustainability of forest management. Master's thesis and Earth Observation, International Institute for Geo-information Science, Enschede, The Netherlands

Lee W-S, Tu W-S (2011) Combined MCDM techniques for exploring company value based on Modigliani–Miller theorem. Expert Syst Appl 38(7):8037–8044

Liao C-N (2011) Fuzzy analytical hierarchy process and multi-segment goal programming applied to new product segmented under price strategy. Comput Ind Eng 61(3):831–841

Liu Q, Shi S, Zhu H, Xiao J (2014) A mutual information-based hybrid feature selection method for software cost estimation using feature clustering. In: 38th annual IEEE computer software and applications conference. IEEE, Vasteras, Sweden, pp 27–32

Liu W, Liu L, Tong F (2017) Least squares support vector machine for ranking solutions of multi-objective water resources allocation optimization models. Water 9:1–15

Liyi M, Shiyu Z, Jian G (2010) A project risk forecast model based on support vector machine. In: IEEE international conference on software engineering and service sciences, Beijing, China, pp 463–466

Madheswaran M, Sivakumar D (2014) Enhancement of prediction accuracy in COCOMO model for software project using neural network. In: International conference on computing, communication and networking technologies. IEEE, Hefei, China, pp 1–5

Marković I, Stojanović M, Božić M, Stanković J (2015) Stock market trend prediction based on the LS-SVM model update algorithm. In: Bogdanova A (ed) ICT innovations 2014. Advances in intelligent systems and computing, vol 311. Springer, Cham, pp 105–114

Marković I, Stojanović M, Stanković J, Stanković M (2017) Stock market trend prediction using ahp and weighted kernel LS-SVM. Soft Comput 21(18):5387–5398

Mendes E, Watson I, Triggs C, Mosley N, Counsell S (2002) A comparison of development effort estimation techniques for Web hypermedia applications. In: Eighth IEEE symposium on software metrics. IEEE, Ottawa, Canada, pp 131–140

Menzies T, Chen Z, Hihn J, Lum K (2006) Selecting best practices for effort estimation. IEEE Trans Softw Eng 32(11):883–895

Menzies T, Caglayan B, He Z, Kocaguneli E, Krall J, Peters F, Turhan B (2012) The promise repository of empirical software engineering data

Mikhailov L, Tsvetinov P (2004) Evaluation of services using a fuzzy analytic hierarchy process. Appl Soft Comput 5(1):23–33

Milios D, Stamelos I, Chatzibagias C (2013) A genetic algorithm approach to global optimization of software cost estimation by analogy. Intell Decis Technol 7(1):45–58

Minku LL, Yao X (2013) Software effort estimation as a multiobjective learning problem. ACM Trans Softw Eng Methodol 22(4):35:1–35:32

Morgenshtern O, Raz T, Dvir D (2007) Factors affecting duration and effort estimation errors in software development projects. Inf Softw Technol 49(8):827–837

Naghadehi MZ, Mikaeil R, Ataei M (2009) The application of fuzzy analytic hierarchy process (FAHP) approach to selection of optimum underground mining method for Jajarm Bauxite Mine, Iran. Expert Syst Appl 36(4):8218–8226

Nisar M, Wang Y-J, Elahi M (2008) Software development effort estimation using fuzzy logic—a survey. In: Fifth international conference on fuzzy systems and knowledge discovery, vol 1. IEEE, Shandong, China, pp 421–427

Oliveira ALI, Braga PL, Lima RMF, Cornélio ML (2010) GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. Inf Softw Technol 52(11):1155–1166

Rastogi H, Dhankhar S, Kakkar M (2014) A survey on software effort estimation techniques. In: Confluence the next generation information technology summit (confluence), 2014 5th international conference. IEEE, Noida, India, pp 826–830

Reddy P, Sudha K, Sree PR, Ramesh S (2010) Software effort estimation using radial basis and generalized regression neural networks. J Comput 2(5):87–92

Ryder J (1998) Fuzzy modeling of software effort prediction. In: Information technology conference. IEEE, Syracuse, USA, pp 53–56

Saaty T (2004) Decision making—the analytic hierarchy and network processes (AHP/ANP). J Syst Sci Syst Eng 13(1):1–35

Saaty TL (2008) Decision making with the analytic hierarchy process. Int J Serv Sci 1(1):83–98

Shaw M (2002) What makes good research in software engineering? Int J Softw Tools Technol Trans 4(1):1–7

Shepperd M, Cartwright M (2001) Predicting with sparse data. IEEE Trans Softw Eng 27(11):987–998

Sheta AF, Rine D, Kassaymeh S (2015) Software effort and function points estimation models based radial basis function and feedforward artificial neural networks. Int J Next Gen Comput 6(3):192–205

Srivastava DK, Chauhan DS, Singh R (2012) VRS model: a model for estimation of efforts and time duration in development of IVR software system. Int J Softw Eng 5(1):27–46

Suykens J, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300

Tang Y-C, Beynon MJ et al (2005) Application and development of a fuzzy analytic hierarchy process within a capital investment study. J Econ Manag 1(2):207–230

Trendowicz A, Münch J, Jeffery R (2008) State of the practice in software effort estimation: a survey and literature review. In: Lin Z, Hu Y, Madachy R, Ravi KV, Boehm BW (eds) Software engineering techniques, vol 4980. Lecture notes in computer science. Springer, Berlin, pp 232–245

Van Laarhoven P, Pedrycz W (1983) A fuzzy extension of Saaty' s priority theory. Fuzzy Sets Syst 11(1–3):229–241

Vapnik V (2013) Nature of statistical learning theory. Information science and statistics, 2nd edn. Springer, New York

Wang Y-M, Luo Y (2009) On rank reversal in decision analysis. Math Comput Model 49(5–6):1221–1229

Wen J, Li S, Lin Z, Hu Y, Huang C (2012) Systematic literature review of machine learning based software development effort estimation models. Inf Softw Technol 54(1):41–59

Wong J, Ho D, Capretz LF (2009) An investigation of using neuro-fuzzy with software size estimation. In: ICSE workshop on software quality. IEEE, Vancouver, Canada, pp 51–58

Xing H-J, Ha MH, Hu BG, Tian DZ (2009) Linear feature-weighted support vector machine. Fuzzy Inf Eng 1(3):289–305

Zadeh LA (1988) Fuzzy logic. Computer 21(4):83–93

Zelkowitz MV, Yeh RT, Hamlet RG, Gannon JD, Basili VR (1984) Software engineering practices in the US and Japan. Computer 17(6):57–70