Research Article

Empirical Study of Homogeneous and Heterogeneous Ensemble Models for Software Development Effort Estimation

Mahmoud O. Elish,¹ Tarek Helmy,^{1,2} and Muhammad Imtiaz Hussain¹

¹ *Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*

² *College of Engineering, Tanta University, Tanta, Egypt*

Correspondence should be addressed to Mahmoud O. Elish; elish@kfupm.edu.sa

Received 13 March 2013; Revised 3 June 2013; Accepted 9 June 2013

Academic Editor: Ren-Jieh Kuo

Copyright © 2013 Mahmoud O. Elish et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate estimation of software development effort is essential for effective management and control of software development projects. Many software effort estimation methods have been proposed in the literature including computational intelligence models. However, none of the existing models proved to be suitable under all circumstances; that is, their performance varies from one dataset to another. The goal of an ensemble model is to manage each of its individual models' strengths and weaknesses automatically, leading to the best possible decision being taken overall. In this paper, we have developed different homogeneous and heterogeneous ensembles of optimized hybrid computational intelligence models for software development effort estimation. Different linear and nonlinear combiners have been used to combine the base hybrid learners. We have conducted an empirical study to evaluate and compare the performance of these ensembles using five popular datasets. The results confirm that individual models are not reliable as their performance is inconsistent and unstable across different datasets. Although none of the ensemble models was consistently the best, many of them were frequently among the best models for each dataset. The homogeneous ensemble of support vector regression (SVR), with the nonlinear combiner adaptive neurofuzzy inference systems-subtractive clustering (ANFIS-SC), was the best model when considering the average rank of each model across the five datasets.

1. Introduction

Software development effort estimation is one of the core tasks in software project management. It is defined as “the process of predicting the effort required to develop a software system” [1]. It is usually measured by the number of person-hours that were spent in developing the software from specification until delivery. The success of a software development project highly depends on accurate estimation of its development effort, among other factors. One of the most common factors of software project failure is inaccurate estimates of needed resources [2]. Overestimation results in wasting of resources, whereas underestimation results in schedule/budget overruns and/or quality compromise. Many software effort estimation methods have been proposed in the literature since 1980s. In recent years, the application of computational intelligence models in estimating software development efforts has been receiving increasing attention

in order to improve the estimation accuracy. However, none of the existing models proved to be suitable under all circumstances. The performance of these models is unreliable, that is, varies from one dataset to another. Therefore, there is a need to build estimation models that are reliable and provide high accuracy. Ensembles of hybrid computational intelligence models are candidates for this objective. The goal of an ensemble model is to manage each of its individual models' strengths and weaknesses automatically, leading to the best possible decision being taken overall.

In this paper, we have developed different homogeneous and heterogeneous ensembles of some optimized hybrid of computational intelligence models for software development effort estimation. Different linear and nonlinear combiners have been used. We have conducted an empirical study to evaluate and compare the performance of these ensembles using five popular datasets. The rest of this paper is organized as follows. Section 2 reviews related work. Section 3

describes the computational intelligence and ensemble models that have been developed. Section 4 reports the conducted empirical study and discusses its results. Section 5 provides concluding remarks and directions for future work.

2. Related Work

Software effort estimation methods can be grouped into three general approaches [3, 4]: expert judgment, algorithmic models, and computational intelligence. Expert judgment makes estimations based on the experience of experts on similar projects. The accuracy of this method greatly depends on the degree in which a new project concurs within the experience and the ability of the expert. According to a recent experiment, it was found that there is a high degree of inconsistency in expert judgment-based estimates of software development effort [5]. Therefore, the process of deriving an estimate is not explicit and thus not repeatable [3]. Jørgensen [6] conducted a review of studies on expert estimation of software development effort.

Algorithmic models represent the relationship between characteristic(s) of a software project, usually software size, and its development effort. These models are parametric in nature with a formula of standard form that is parameterized from historical data. Examples of such models include constructive cost model (COCOMO) [7], function points analysis [8], and software lifecycle management (SLIM) [9]. Algorithmic models are unable to capture the complex set of relationships. Moreover, they need to be calibrated or adjusted to local circumstances [4, 10].

Computational intelligence models, in recent years, have been widely applied to software effort estimation. Examples include neural networks [4, 11–13], Bayesian network [14], fuzzy logic [3], regression trees (RT) [15, 16], case-based reasoning [10, 17, 18], genetic programming [19, 20], and support vector regression [15, 21]. Some advantages of computational intelligence models include their ability to model the complex set of relationships between effort and its drivers and their ability to learn from historical projects data [10]. Wen et al. [1] performed a systematic literature review of software development effort estimation based on computational intelligence models.

Recently, few research studies have investigated the use of homogeneous ensemble models for software effort estimation. Braga et al. [22] investigated the use of bagging predictors for estimation of software project effort using two versions of a NASA dataset. In particular, they applied bagging to linear regression, multilayer perceptron (MLP), M5P regression trees, M5P model trees, and support vector regression (SVR). They concluded that bagging was able to improve the performance of all models except SVR. Kultur et al. [23, 24] proposed ensemble of neural networks with associative memory (ENNA) for estimating software development effort. As neural network model, they used MLP and training sets were generated by bootstrapping. The outputs of the base learners were combined by taking the average of largest cluster obtained using adaptive resonance theory algorithm. The results showed that ENNA are significantly better than neural

network in terms of accuracy and robustness. Minku and Yao [25, 26] investigated bagging ensemble with MLPs, with Radial Basis Function (RBF) network, and with regression trees. In addition, they investigated random ensemble with MLPs and negative correlation learning ensemble with MLPs. The outputs of the base learners were combined by taking the average. They observed that bagging ensemble of regression trees performed well in comparison to other approaches.

Some other few studies have recently investigated the use of heterogeneous ensemble models for software effort estimation. Kocaguneli et al. [27] evaluated a heterogeneous ensemble of multiple learners by averaging across them. However, no improvement of the estimation accuracy of software effort was achieved. Kocaguneli et al. [28] evaluated ensembles of preprocessed estimation methods. They used simple linear combination schemes, which are the mean, median, and inverse-ranked weighted mean (IRWM). Elish [29] evaluated the extent to which the voting ensemble model, with median combination rule, offers reliable and improved estimation accuracy over five individual models: MLP, RBF, RT, K-nearest neighbor (KNN), and SVR in estimating software development effort. In three out of the five datasets that were used in that study, the ensemble model outperformed the individual models. In the other two datasets, the ensemble model achieved the second best performance.

This paper differs from the above related works on the use of ensemble models for software effort estimation in several aspects. This paper investigates and compares both homogeneous and heterogeneous ensembles of hybrid computational intelligence models. Furthermore, in addition to simple linear combiners, this paper investigates and compares several nonlinear combiners. A comparison between this paper and related works is provided in Table 1.

3. Hybrid Computational Intelligence and Ensemble Models

A hybrid computational intelligent (HCI) model combines at least two computational intelligent (CI) techniques. For example, the combination of an artificial neural network (ANN) with a fuzzy inference system (FIS) results in a hybrid neurofuzzy system. HCI models are defined as any effective combination of CI techniques in sequential or parallel manner that perform superior to simple CI techniques [30]. The main challenge of HCI model is the collaboration efficiency of each component. Another important factor of an HCI model is the speed of process and the time needed to produce a generalized high-performance decision model.

In this paper we have used nonlinear (categorical) principal component analysis (PCA) along with the different CI models results in HCI models. PCA was first introduced by Pearson in 1901 and become a popular tool in data analysis. PCA finds the directions in which a cloud of data points is stretched most. The objective of PCA is to perform dimensionality reduction while preserving the randomness in the high-dimensional space. PCA performs a mapping of the data to a lower dimensional space in such a way that the

TABLE 1: Comparison of related works on the use of ensemble models for software effort estimation.

Study	Ensemble type	Base learner(s)	Combination rule(s)	Number of datasets
Braga et al. [22]	Homogeneous (bagging)	Linear regression	Linear (averaging)	1
	Homogeneous (bagging)	MLP	Linear (averaging)	
	Homogeneous (bagging)	M5P regression trees	Linear (averaging)	
	Homogeneous (bagging)	M5P model trees	Linear (averaging)	
	Homogeneous (bagging)	SVR	Linear (averaging)	
Kultur et al. [23, 24]	Homogeneous (bootstrapping)	MLP	Nonlinear (average of largest cluster obtained using adaptive resonance theory (ART) algorithm)	5
Minku and Yao [25, 26]	Homogeneous (bagging, random, negative correlation learning)	MLP	Linear (averaging)	18
	Homogeneous (bagging)	RBF	Linear (averaging)	
	Homogeneous (bagging)	Regression Trees	Linear (averaging)	
Kocaguneli et al. [27]	Heterogeneous	Gaussian process, MLP, RBF, SMOReg, SVMReg, IBk, LWL, additive regression, bagging with decision tree, RandomSubSpace, DecisionStump, M5P, ConjunctiveRule, DecisionTable	Linear (averaging)	3
Kocaguneli et al. [28]	Heterogeneous	ABE0-INN, ABE0-5NN, SWReg, CART (yes), CART (no), NNet, LReg, PCR, PLSR	Linear (mean, median, inverse-ranked weighted mean (IRWM))	20
Elish [29]	Heterogeneous	MLP, RBF, RT, KNN, SVR	Linear (median)	5
This paper	Homogeneous (bagging)	MLP	Linear (averaging, weighted averaging) and nonlinear (MLP SVR, FIS-FCM, FIS-SC, ANFIS-FCM, ANFIS-SC)	5
	Homogeneous (bagging)	SVR		
	Homogeneous (bagging)	ANFIS		
	Heterogeneous	MLP, SVR, ANFIS		

variance of the data in the low-dimensional representation is maximized. The basic idea behind using PCA for feature selection prior to regression is to select variables according to the magnitude (from largest to smallest in absolute values) of their coefficients. In the proposed ensemble models, PCA seeks to replace more or less correlated variables by uncorrelated combinations (projections) of the original variables. Also, PCA is used to perform dimension reduction and variable selection based on the resulting variable loadings.

3.1. The Ensemble Model. An ensemble model employs a group of multiple learning algorithms and combines their outputs acting as a single decision maker. Figure 1 shows an ensemble of N number of CI models. Every CI model has limitations and so different learning algorithm suit different

problems. The principle of ensemble is that the combined results of ensemble model should have better overall accuracy than any individual member. Numerous studies showed that ensemble accuracy significantly exceeds the single model [31–33]. The goal of ensemble model is to manage each individual model's strengths and weaknesses automatically, leading to the best possible decision being taken overall. In fact, one of the motivations of using the ensemble approach is to avoid overfitting that is caused by high variance. Ensemble models take a combination of several hypotheses, which tends to cancel out overfitting errors. Variance reduction methods such as bagging can help and typically that has been done. Indeed, best results are often obtained by bagging overfitted classifiers. Typically, the proposed ensembles are constructed in two steps. First, a number of base learners are produced, which can be generated in a parallel style

(bagging) or in a sequential style (boosting) where the generation of a base learner has influence on the generation of subsequent learners. Then, the base learners are combined to use, where among the most popular combination schemes are *majority voting* for classification and *weighted averaging* for regression. Simple averaging often provides an effective method of combining continuous outputs from individual CI models using $O_{en} = (1/n) \sum_{i=1}^n O_i$, where O_i ($i = 1, 2, \dots, n$) is the output of CI member i and O_{en} is the combined output. The O_{en} formula assumes that outputs from all CI models are of equal weight. In practice, some outputs may have greater weights than others, and individual CI outputs may be weighted to reflect this fact. The combined output is then a weighted sum of the outputs of all members as given by $O_{en} = (1/n) \sum_{i=1}^n \alpha_i O_i$, where $\sum_{i=1}^n \alpha_i = 1$. To assign weight of the members of ensemble model we predicted the whole training data to measure each member's performance in terms of RMSE and assigned highest weight to the model having lowest RMSE.

The formula for weighted average method is

$$\frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}, \quad (1)$$

where the weight calculation formula is

$$W_i = \frac{\sum_{i=1}^n \text{RMSE}_i - \text{RMSE}_i}{(n-1) \times \sum_{i=1}^n \text{RMSE}_i}. \quad (2)$$

To combine the outputs of the ensemble, we have used linear and nonlinear approaches. For linear method we have used averaging and weighted averaging methods. We have used the CI models as nonlinear combiner. We have chosen the CI models such as ANN, SVR, FIS, and adaptive neurofuzzy inference system (ANFIS) created with fuzzy C-means clustering (FCM) and subtractive clustering (SC) as a combiner. For ANN combiner we have used MLP with two neurons in the hidden layer with log-sigmoidal activation function, and in the output layer we have used tan-sigmoidal activation function and we have used either of the trainlm or trainrp training algorithms. For SVR we have used "Gaussian" type kernel with γ value 0.5. The other parameters are, for example, $C = 0.5$, $\lambda = 1e-7$, and $\epsilon = 0.0001$. For combining using FIS, we have used FCM with 6 clusters and radius of 0.3 for SC.

3.2. Heterogeneous and Homogeneous Ensembles. Heterogeneous ensemble consists of members having different base learning algorithms. We developed one heterogeneous ensemble model having PCA-based CI models of type MLP, SVR, and ANFIS. At first we provided the input in MLP. We selected the poorly predicted training data by MLP and provided it to train the SVR and later on the poorly predicted training data by SVR is provided to ANFIS for training. In this way the model would become diverse by having training datasets.

Homogeneous ensemble consists of members having a single-type base learning algorithm. In this case ensemble members can be different by the structure. We developed

FIGURE 1: Ensemble model structure.

three homogeneous ensemble models and each has three PCA-based CI models of type MLP, SVR, and ANFIS. Moreover, in the proposed ensembles, we optimized their parameters using an evolutionary algorithm based on the genetic algorithm (GA). To improve the efficiency of PCA approach, the GA has been used to select the features that would increase the performance in both training phase and test phase. We used GA to extract the most important feature for improving time and accuracy of their methods, and the PCA is used for feature extraction and classification, respectively.

Table 2 lists the homogeneous and heterogeneous ensembles that have been investigated in this paper. A total of 32 different ensemble models have been evaluated and compared. For each ensemble, the table provides its base learner(s), combination type, and rule. The naming convention (abbreviation) for these ensemble models is as follows: EnsembleType-BaseLearner(s)-[CombinationRule]. EnsembleType is either HM (homogeneous) or HT (heterogeneous). BaseLearner(s) is either MLP, or SVR, or ANFIS for HM ensembles; or MLP, SVR, and ANFIS for HT ensembles. There are eight combination rules: two linear (averaging and weighted averaging) and six nonlinear (MLP, SVR, FIS-FCM, FIS-SC, ANFIS-FCM, and ANFIS-SC).

3.2.1. Selecting Training Set for Each Ensemble Member. At first we divided the whole datasets into training and testing datasets. Around 80% of the datasets is used for training and the rest 20% were used for testing. The ensemble members are actually trained using 80% data of the training set and the rest is used for model validation. After the first run of the algorithm, in each of the following runs we have selected the same amount of actual training data, that is, the 80% of the whole training set as selected in the previous run which are poorly predicted by the CI model.

4. Empirical Evaluation

We conducted an empirical study to evaluate and compare the performance of the homogeneous and heterogeneous ensemble models under investigation in estimating software development effort. This section discusses the conducted empirical study and its results.

TABLE 2: Investigated homogeneous and heterogeneous ensemble models.

Ensemble type	Base learner(s)	Combination type	Combination rule	Abbreviation
Homogeneous	MLP	Linear	Averaging	HM-MLP-[Avg]
			Weighted averaging	HM-MLP-[WtAvg]
		Nonlinear	MLP	HM-MLP-[MLP]
			SVR	HM-MLP-[SVR]
			FIS-FCM	HM-MLP-[FIS-FCM]
			FIS-SC	HM-MLP-[FIS-SC]
			ANFIS-FCM	HM-MLP-[ANFIS-FCM]
			ANFIS-SC	HM-MLP-[ANFIS-SC]
	SVR	Linear	Averaging	HM-SVR-[Avg]
			Weighted averaging	HM-SVR-[WtAvg]
		Nonlinear	MLP	HM-SVR-[MLP]
			SVR	HM-SVR-[SVR]
			FIS-FCM	HM-SVR-[FIS-FCM]
			FIS-SC	HM-SVR-[FIS-SC]
			ANFIS-FCM	HM-SVR-[ANFIS-FCM]
			ANFIS-SC	HM-SVR-[ANFIS-SC]
	ANFIS	Linear	Averaging	HM-ANFIS-[Avg]
			Weighted averaging	HM-ANFIS-[WtAvg]
		Nonlinear	MLP	HM-ANFIS-[MLP]
			SVR	HM-ANFIS-[SVR]
			FIS-FCM	HM-ANFIS-[FIS-FCM]
			FIS-SC	HM-ANFIS-[FIS-SC]
			ANFIS-FCM	HM-ANFIS-[ANFIS-FCM]
			ANFIS-SC	HM-ANFIS-[ANFIS-SC]
Heterogeneous	MLP, SVR, and ANFIS	Linear	Averaging	HT-(MLP, SVR, ANFIS)-[Avg]
			Weighted averaging	HT-(MLP, SVR, ANFIS)-[WtAvg]
			MLP	HT-(MLP, SVR, ANFIS)-[MLP]
			SVR	HT-(MLP, SVR, ANFIS)-[SVR]
		Nonlinear	FIS-FCM	HT-(MLP, SVR, ANFIS)-[FIS-FCM]
			FIS-SC	HT-(MLP, SVR, ANFIS)-[FIS-SC]
			ANFIS-FCM	HT-(MLP, SVR, ANFIS)-[ANFIS-FCM]
			ANFIS-SC	HT-(MLP, SVR, ANFIS)-[ANFIS-SC]

4.1. Datasets. Five well-known datasets were used in this empirical study. These datasets, which are described next, have been widely used in the literature. A summary of their characteristics is provided in Table 3.

4.1.1. Albrecht Dataset. Albrecht dataset [8] consists of 24 software projects developed by the IBM DP Services organization. It has six independent (input) variables: “input count,” “output count,” “query count,” “file count,” “function points,” and “source lines of code.” The dependent (output) variable is the software development effort measured by the number of work hours required to design, develop, and test each application.

4.1.2. Miyazaki Dataset. Miyazaki dataset [34] consists of 48 software projects collected by Fujitsu Large Systems

Users Group. It has seven independent variables: “number of screens,” “number of forms,” “number of files,” “number of data elements in screens,” “number of data elements in forms,” “number of data elements in files,” and “source lines of code.” The dependent variable is the software development effort measured by the number of man-months from systems design to systems test spent for each system, including indirect effort such as project management. A man-month equals 160 hours of working time.

4.1.3. Maxwell Dataset. Maxwell dataset [35] consists of 62 software projects from one of the biggest commercial banks in Finland. It has 25 independent variables that measure various software characteristics such as application type, and size. The dependent variable is the software development effort

TABLE 3: Characteristics of datasets.

Dataset	No. of observations (size)	No. of features (Input variables)	No. of numerical features	No. of categorical features
Albrecht	24	6	6	0
Miyazaki	48	7	7	0
Maxwell	62	25	3	22
COCOMO	63	16	16	0
Desharnais	77	8	7	1

TABLE 4: Performance of individual and hybrid models based on EF metric.

		Dataset				
		Albrecht	Miyazaki	Maxwell	COCOMO	Desharnais
SVR	Individual	0.243	0.274	0.326	0.010	0.445
	PCA	0.418	0.503	0.478	0.081	0.258
	PCA-GA	0.000	1.307	0.538	0.070	0.062
MLP	Individual	0.610	0.299	0.330	0.092	0.252
	PCA	0.300	0.499	0.234	0.275	0.227
	PCA-GA	0.245	0.154	0.193	0.017	0.070
ANFIS	Individual	0.251	0.538	0.087	0.043	0.187
	PCA	0.139	1.305	0.110	0.050	0.000
	PCA-GA	0.000	0.203	0.059	0.007	0.000

measured by the number of hours of the work carried out by the software supplier from specification until delivery.

4.1.4. COCOMO Dataset. COCOMO dataset [7] consists of 63 software projects including business, scientific, systems, real-time, and support software projects. It has 16 independent variables that measure product, project, computer, and personnel attributes. The dependent variable is the software development effort measured in person hours.

4.1.5. Desharnais Dataset. Desharnais dataset [36] consists of 77 software projects from a Canadian Software house. It has eight independent variables: “team experience,” “manager’s experience,” “length of project,” “transactions,” “entities,” “adjusted function points,” “development environment,” and “programming language.” The dependent variable is the software development effort measured in person hours.

4.2. Performance Evaluation Metrics. In order to assess and compare the different estimation models, three performance evaluation metrics were considered. The first metric is mean magnitude of relative error (MMRE), which is calculated as follows:

$$\text{MMRE} = \frac{1}{N} \sum_{i=1}^N \frac{|x_i - \hat{x}_i|}{x_i}, \quad (3)$$

where x_i and $\hat{x}_i$ are the actual and estimated values of observation i , respectively, in a dataset of N observations. The second metric is PRED(25), which is a measure of the percentage of observations whose magnitude of relative error (MRE) is less than 0.25. A good estimation model will minimize MMRE and maximize PRED(25).

The third performance metric is a recently proposed evaluation function (EF) [37], which is a combination of MMRE and PRED(25). It is calculated as follows:

$$\text{EF} = \frac{\text{PRED}(25)}{1 + \text{MMRE}}. \quad (4)$$

4.3. Evaluation of Hybrid Models. This section evaluates whether the hybridization of an individual model improves its estimation performance. If so, we will use the hybrid version of it in the development of the ensemble models; otherwise we use it as it is. In other words, the performance of the individual SVR model was compared to the hybrid PCA-SVR model and the hybrid PCA-GA-SVR model. The individual MLP and ANFIS models were also compared against their hybrid versions. Table 4 reports the performance of the individual and hybrid models based on the EF metric in each of the five datasets. Figures 2–6 show histograms of the models’ performance for the five datasets, respectively.

In Albrecht dataset, as observed from Table 4 and Figure 2, the hybrid PCA-SVR model performed better than both the individual SVR model and the hybrid PCA-GA-SVR model. However, the individual MLP model performed better than both the hybrid PCA-MLP model and the hybrid PCA-GA-MLP model. The individual ANFIS model also performed better than its hybrid versions. For this dataset, we accordingly developed homogeneous and heterogeneous ensembles of hybrid PCA-SVR model and individual MLP and ANFIS models.

In Miyazaki dataset, as observed from Table 4 and Figure 3, the hybrid PCA-GA-SVR model performed better than both the individual SVR model and the hybrid PCA-SVR model. The hybrid PCA-MLP model performed better

FIGURE 2: Performance of individual and hybrid models based on EF metric using Albrecht dataset.

than both the individual MLP model and the hybrid PCA-GA-MLP model. Similarly, the hybrid PCA-ANFIS model performed better than both the individual ANFIS model and the hybrid PCA-GA-ANFIS model. For this dataset, we accordingly developed homogeneous and heterogeneous ensembles of the hybrid PCA-GA-SVR model, the hybrid PCA-MLP model, and the hybrid PCA-ANFIS model.

In Maxwell dataset, as observed from Table 4 and Figure 4, the hybrid PCA-GA-SVR model performed better than both the individual SVR model and the hybrid PCA-SVR model. The individual MLP model performed better than the hybrid versions. The hybrid PCA-ANFIS model performed better than both the individual ANFIS model and the hybrid PCA-GA-ANFIS model. For this dataset, we accordingly developed homogeneous and heterogeneous ensembles of the hybrid PCA-GA-SVR model, the individual MLP model, and the hybrid PCA-ANFIS model.

In COCOMO dataset, as observed from Table 4 and Figure 5, the hybrid PCA-SVR model performed better than both the individual SVR model and the hybrid PCA-GA-SVR model. Similarly, the hybrid PCA-MLP model performed better than both the individual MLP model and the hybrid PCA-GA-MLP model, and the hybrid PCA-ANFIS model performed better than both the individual ANFIS model and the hybrid PCA-GA-ANFIS model. For this dataset, we accordingly developed homogeneous and heterogeneous ensembles of hybrid PCA version of the individual models.

In Desharnais dataset, as observed from Table 4 and Figure 6, the individual SVR model performed better than both the hybrid PCA-SVR model and the hybrid PCA-GA-SVR model. Similarly, the individual MLP and ANFIS models performed better than their hybrid models. For this dataset, we accordingly developed homogeneous and heterogeneous ensembles of individual models rather than using their hybrid versions.

FIGURE 3: Performance of individual and hybrid models based on EF metric using Miyazaki dataset.

FIGURE 4: Performance of individual and hybrid models based on EF metric using Maxwell dataset.

4.4. Evaluation of Homogeneous and Heterogeneous Ensemble Models. This section evaluates and compares the estimation performance of the homogeneous and heterogeneous ensemble models under investigation. Tables 4, 5, 6, 7, 8 and 9 report the performance of the individual models, the homogeneous ensemble models, and the heterogeneous ensemble models in estimating software development effort using Albrecht, Miyazaki, Maxwell, COCOMO, and Desharnais datasets, respectively. Figures 7, 9, 11, 13, and 15 show histograms of the EF measure for each model's performance in each of the five datasets, respectively. Figures 8, 10, 12, 14, and 16 show five plots of the MMRE versus the PRED(25) values that were achieved by each model in each dataset, respectively. Relatively accurate estimation models appear in the top left

FIGURE 5: Performance of individual and hybrid models based on EF metric using COCOMO dataset.

FIGURE 6: Performance of individual and hybrid models based on EF metric using Desharnais dataset.

corner of these plots. The following subsections discuss the results based on each dataset and then the overall results based on all datasets.

4.4.1. Results Based on Albrecht Dataset. Among the individual models, the MLP model achieved the best performance in terms of MMRE, PRED(25), and EF. By comparing the performance of the homogeneous ensembles of MLP, it can be observed that HM-MLP-[WtAvg], and HM-MLP-[FIC-SC] were the best models. Moreover, only three homogeneous ensembles of MLP (i.e., HM-MLP-[WtAvg], HM-MLP-[FIC-SC] and HM-MLP-[MLP]) achieved better EF than the individual MLP model, whereas the other ensembles of MLP were worse than it. By comparing the performance of the

homogeneous ensembles of SVR, it can be noticed that HM-SVR-[ANFIS-SC] was the best, followed by the HM-SVR-[ANFIS-FCM]. Furthermore, all homogeneous ensembles of SVR except HM-SVR-[FIC-SC] improved the performance of the individual SVR model in terms of EF. By comparing the performance of the homogeneous ensembles of ANFIS, it can be observed that HM-ANFIS-[MLP] was the best among them in terms of EF. In addition, only three homogeneous ensembles of ANFIS (i.e., HM-ANFIS-[Avg], HM-ANFIS-[FIC-SC], and HM-ANFIS-[ANFIS-SC]) achieved worse EF than the individual ANFIS model, whereas the other ensembles of ANFIS were better than it.

Among the heterogeneous ensemble models, HT-(MLP, -SVR, ANFIS)-[ANFIS-FCM] and HT-(MLP, SVR, ANFIS)-[ANFIS-SC] achieved relatively better performance than the other heterogeneous ensembles. It is interesting to observe that the individual MLP model performed better than all the heterogeneous ensembles. The distribution of the top 10 models, in terms of EF, is as follows: 1 individual model (MLP), 5 ensembles of MLP, 3 ensembles of SVR, and 1 ensemble of ANFIS. None of the heterogeneous ensembles was among the top 10 models.

4.4.2. Results Based on Miyazaki Dataset. Among the individual models, the ANFIS model achieved the best performance in terms of MMRE and EF, and the SVR model was the best in terms of PRED(25). By comparing the performance of the homogeneous ensembles of MLP, it can be observed that the HM-MLP-[ANFIS-FCM] model was the best. Moreover, only three homogeneous ensembles of MLP (i.e., HM-MLP-[MLP], HM-MLP-[FIC-SC], and HM-MLP-[ANFIS-FCM]) achieved better EF than the individual MLP model, whereas the other ensembles of MLP were worse than it. By comparing the performance of the homogeneous ensembles of SVR, it can be noticed that HM-SVR-[ANFIS-SC] was the best. Furthermore, all homogeneous ensembles of SVR except HM-SVR-[SVR] improved the performance of the individual SVR model in terms of EF. By comparing the performance of the homogeneous ensembles of ANFIS, it can be observed that HM-ANFIS-[Avg] and HM-ANFIS-[WtAvg] were the best models among them in terms of EF. In addition, only three homogeneous ensembles of ANFIS (i.e., HM-ANFIS-[Avg], HM-ANFIS-[WtAvg], and HM-ANFIS-[ANFIS-FCM]) achieved better EF than the individual ANFIS model, whereas the other ensembles of ANFIS were worse than it.

Among the heterogeneous ensemble models, HT-(MLP, SVR, ANFIS)-[Avg], HT-(MLP, SVR, ANFIS)-[WtAvg], and HT-(MLP, SVR, ANFIS)-[MLP] achieved relatively better performance than the other heterogeneous ensembles. The distribution of the top 10 models, in terms of EF, is as follows: 1 ensemble of MLP, 3 ensembles of SVR, 3 ensembles of ANFIS, and 3 heterogeneous ensembles. None of the individual model was among the top 10 models.

4.4.3. Results Based on Maxwell Dataset. Among the individual models, the MLP model achieved the best performance in terms of MMRE, whereas the SVR model was the best

FIGURE 7: Histograms of models' EF metric using Albrecht dataset.

FIGURE 8: MMRE versus PRED(25) by each model using Albrecht dataset.

FIGURE 9: Histograms of models' EF metric using Miyazaki dataset.

FIGURE 10: MMRE versus PRED(25) by each model using Miyazaki dataset.

TABLE 5: Models' performance using Albrecht dataset.

Model type	Model	MMRE	PRED(25)	EF
Individual	MLP	64.60	40.0	0.61
	SVR	81.40	20.0	0.24
	ANFIS	78.61	20.0	0.25
Homogeneous ensemble (MLP)	HM-MLP-[Avg]	32.99	20.0	0.59
	HM-MLP-[WtAvg]	44.13	40.0	0.89
	HM-MLP-[MLP]	54.23	40.0	0.72
	HM-MLP-[SVR]	74.38	40.0	0.53
	HM-MLP-[FIS-FCM]	61.43	20.0	0.32
	HM-MLP-[FIS-SC]	43.87	40.0	0.89
	HM-MLP-[ANFIS-FCM]	72.52	40.0	0.54
	HM-MLP-[ANFIS-SC]	63.04	20.0	0.31
Homogeneous ensemble (SVR)	HM-SVR-[Avg]	67.65	20.0	0.29
	HM-SVR-[WtAvg]	68.02	20.0	0.29
	HM-SVR-[MLP]	69.30	20.0	0.28
	HM-SVR-[SVR]	74.25	40.0	0.53
	HM-SVR-[FIS-FCM]	77.52	20.0	0.25
	HM-SVR-[FIS-SC]	94.58	0.0	0.00
	HM-SVR-[ANFIS-FCM]	48.63	60.0	1.21
	HM-SVR-[ANFIS-SC]	38.20	60.0	1.53
Homogeneous ensemble (ANFIS)	HM-ANFIS-[Avg]	75.88	0.0	0.00
	HM-ANFIS-[WtAvg]	69.43	20.0	0.28
	HM-ANFIS-[MLP]	68.16	40.0	0.58
	HM-ANFIS-[SVR]	80.83	40.0	0.49
	HM-ANFIS-[FIS-FCM]	67.24	20.0	0.29
	HM-ANFIS-[FIS-SC]	108.83	0.0	0.00
	HM-ANFIS-[ANFIS-FCM]	67.30	20.0	0.29
	HM-ANFIS-[ANFIS-SC]	136.01	0.0	0.00
Heterogeneous ensemble (MLP, SVR, ANFIS)	HT-(MLP, SVR, ANFIS)-[Avg]	84.79	0.0	0.00
	HT-(MLP, SVR, ANFIS)-[WtAvg]	73.30	20.0	0.27
	HT-(MLP, SVR, ANFIS)-[MLP]	92.03	20.0	0.21
	HT-(MLP, SVR, ANFIS)-[SVR]	87.70	20.0	0.23
	HT-(MLP, SVR, ANFIS)-[FIS-FCM]	106.82	0.0	0.00
	HT-(MLP, SVR, ANFIS)-[FIS-SC]	86.77	0.0	0.00
	HT-(MLP, SVR, ANFIS)-[ANFIS-FCM]	60.59	20.0	0.32
	HT-(MLP, SVR, ANFIS)-[ANFIS-SC]	58.82	20.0	0.33

in terms of PRED(25). Both models achieved the best EF value. By comparing the performance of the homogeneous ensembles of MLP, it can be observed that HM-MLP-[ANFIS-SC] was the best model based on PRED(25) and EF metrics. Moreover, only three homogeneous ensembles of MLP (i.e., HM-MLP-[Avg], HM-MLP-[FIS-SC], and HM-MLP-[ANFIS-SC]) achieved better EF than the individual MLP model, whereas the other ensembles of MLP were worse than it. By comparing the performance of the homogeneous ensembles of SVR, it can be noticed that HM-SVR-[MLP] was the best, followed by HM-SVR-[ANFIS-SC] in terms of EF. Furthermore, all other homogeneous ensembles of SVR performed worse than the individual SVR model in terms

of EF. By comparing the performance of the homogeneous ensembles of ANFIS, it can be observed that HM-ANFIS-[ANFIS-SC] was the best among them in terms of MMRE, PRED(25), and EF. In addition, only two homogeneous ensembles of ANFIS (i.e., HM-ANFIS-[FIS-FCM] and HM-ANFIS-[ANFIS-FCM]) achieved worse EF than the individual ANFIS model, whereas the other ensembles of ANFIS were better than it.

Among the heterogeneous ensemble models, HT-(MLP, SVR, ANFIS)-[FIS-SC] and HT-(MLP, SVR, ANFIS)-[ANFIS-SC] achieved relatively better performance than the other heterogeneous ensembles. The distribution of the top 10 models, in terms of EF, is as follows: 3 ensembles of

TABLE 6: Models' performance using Miyazaki dataset.

Model type	Model	MMRE	PRED(25)	EF
Individual	MLP	65.86	20.0	0.30
	SVR	108.37	30.0	0.27
	ANFIS	36.18	20.0	0.54
Homogeneous ensemble (MLP)	HM-MLP-[Avg]	176.62	20.0	0.11
	HM-MLP-[WtAvg]	176.60	20.0	0.11
	HM-MLP-[MLP]	69.23	30.0	0.43
	HM-MLP-[SVR]	107.68	30.0	0.28
	HM-MLP-[FIS-FCM]	317.26	10.0	0.03
	HM-MLP-[FIS-SC]	82.92	40.0	0.48
	HM-MLP-[ANFIS-FCM]	56.14	40.0	0.70
	HM-MLP-[ANFIS-SC]	95.54	10.0	0.10
Homogeneous ensemble (SVR)	HM-SVR-[Avg]	48.05	30.0	0.61
	HM-SVR-[WtAvg]	49.49	30.0	0.59
	HM-SVR-[MLP]	93.79	40.0	0.42
	HM-SVR-[SVR]	105.06	20.0	0.19
	HM-SVR-[FIS-FCM]	85.07	40.0	0.46
	HM-SVR-[FIS-SC]	39.89	40.0	0.98
	HM-SVR-[ANFIS-FCM]	106.68	30.0	0.28
	HM-SVR-[ANFIS-SC]	35.49	60.0	1.64
Homogeneous ensemble (ANFIS)	HM-ANFIS-[Avg]	30.74	30.0	0.95
	HM-ANFIS-[WtAvg]	30.74	30.0	0.95
	HM-ANFIS-[MLP]	103.44	30.0	0.29
	HM-ANFIS-[SVR]	114.64	40.0	0.35
	HM-ANFIS-[FIS-FCM]	138.52	20.0	0.14
	HM-ANFIS-[FIS-SC]	171.95	20.0	0.12
	HM-ANFIS-[ANFIS-FCM]	101.49	70.0	0.68
	HM-ANFIS-[ANFIS-SC]	118.47	10.0	0.08
Heterogeneous ensemble (MLP, SVR, ANFIS)	HT-(MLP, SVR, ANFIS)-[Avg]	58.29	50.0	0.84
	HT-(MLP, SVR, ANFIS)-[WtAvg]	50.84	50.0	0.96
	HT-(MLP, SVR, ANFIS)-[MLP]	83.82	60.0	0.71
	HT-(MLP, SVR, ANFIS)-[SVR]	114.18	30.0	0.26
	HT-(MLP, SVR, ANFIS)-[FIS-FCM]	130.00	20.0	0.15
	HT-(MLP, SVR, ANFIS)-[FIS-SC]	78.71	30.0	0.38
	HT-(MLP, SVR, ANFIS)-[ANFIS-FCM]	149.92	20.0	0.13
	HT-(MLP, SVR, ANFIS)-[ANFIS-SC]	64.95	30.0	0.45

MLP, 2 ensembles of SVR, 3 ensembles of ANFIS, and 2 heterogeneous ensembles. None of the individual models was among the top 10 models.

4.4.4. Results Based on COCOMO Dataset. Among the individual models, the MLP model achieved the best performance in terms of MMRE, PRED(25), and EF. By comparing the performance of the homogeneous ensembles of MLP, it can be observed that HM-MLP-[Avg], HM-MLP-[WtAvg], and HM-MLP-[MLP] were the best models. Moreover, only these three homogeneous ensembles of MLP, in addition to

HM-MLP-[ANFIS-FCM], achieved better EF than the individual MLP model. The other ensembles of MLP were worse than it. By comparing the performance of the homogeneous ensembles of SVR, it can be noticed that the HM-SVR-[MLP] was the best. Furthermore, only three homogeneous ensembles of SVR (i.e., HM-SVR-[SVR], HM-SVR-[FIS-FCM], and HM-SVR-[FIS-SC]) achieved worse EF than the individual SVR model, whereas the other ensembles of SVR were better than it. By comparing the performance of the homogeneous ensembles of ANFIS, it can be observed that the HM-ANFIS-[ANFIS-SC] was the best among them in terms of EF. In addition, only three homogeneous ensembles

FIGURE 11: Histograms of models' EF metric using Maxwell dataset.

FIGURE 12: MMRE versus PRED(25) by each model using Maxwell dataset.

TABLE 7: Models' performance using Maxwell dataset.

Model type	Model	MMRE	PRED(25)	EF
Individual	MLP	74.80	25.0	0.33
	SVR	101.31	33.3	0.33
	ANFIS	190.49	16.7	0.09
Homogeneous ensemble (MLP)	HM-MLP-[Avg]	53.24	25.0	0.46
	HM-MLP-[WtAvg]	66.03	16.7	0.25
	HM-MLP-[MLP]	109.23	8.3	0.08
	HM-MLP-[SVR]	141.49	16.7	0.12
	HM-MLP-[FIS-FCM]	82.56	16.7	0.20
	HM-MLP-[FIS-SC]	64.36	25.0	0.38
	HM-MLP-[ANFIS-FCM]	2969.17	33.3	0.01
	HM-MLP-[ANFIS-SC]	60.53	33.3	0.54
Homogeneous ensemble (SVR)	HM-SVR-[Avg]	77.74	25.0	0.32
	HM-SVR-[WtAvg]	100.87	16.7	0.16
	HM-SVR-[MLP]	85.26	33.3	0.39
	HM-SVR-[SVR]	139.33	16.7	0.12
	HM-SVR-[FIS-FCM]	106.43	16.7	0.16
	HM-SVR-[FIS-SC]	78.92	25.0	0.31
	HM-SVR-[ANFIS-FCM]	43183.76	0.0	0.00
	HM-SVR-[ANFIS-SC]	66.46	25.0	0.37
Homogeneous ensemble (ANFIS)	HM-ANFIS-[Avg]	64.06	25.0	0.38
	HM-ANFIS-[WtAvg]	65.88	25.0	0.37
	HM-ANFIS-[MLP]	92.25	8.3	0.09
	HM-ANFIS-[SVR]	139.34	16.7	0.12
	HM-ANFIS-[FIS-FCM]	119.37	8.3	0.07
	HM-ANFIS-[FIS-SC]	75.25	25.0	0.33
	HM-ANFIS-[ANFIS-FCM]	419245.17	8.3	0.00
	HM-ANFIS-[ANFIS-SC]	59.89	25.0	0.41
Heterogeneous ensemble (MLP, SVR, ANFIS)	HT-(MLP, SVR, ANFIS)-[Avg]	63.31	16.7	0.26
	HT-(MLP, SVR, ANFIS)-[WtAvg]	63.79	16.7	0.26
	HT-(MLP, SVR, ANFIS)-[MLP]	96.19	8.3	0.09
	HT-(MLP, SVR, ANFIS)-[SVR]	167.08	16.7	0.10
	HT-(MLP, SVR, ANFIS)-[FIS-FCM]	114.27	0.0	0.00
	HT-(MLP, SVR, ANFIS)-[FIS-SC]	62.34	25.0	0.39
	HT-(MLP, SVR, ANFIS)-[ANFIS-FCM]	312667.24	8.3	0.00
	HT-(MLP, SVR, ANFIS)-[ANFIS-SC]	59.49	25.0	0.41

of ANFIS (i.e., HM-ANFIS-[Avg], HM-ANFIS-[WtAvg], and HM-ANFIS-[ANFIS-SC]) achieved better EF than the individual ANFIS model, whereas the other ensembles of ANFIS were worse than it.

Among the heterogeneous ensemble models, HT-(MLP, SVR, ANFIS)-[WtAvg] and HT-(MLP, SVR, ANFIS)-[ANFIS-SC] achieved relatively better performance than the other heterogeneous ensembles. The distribution of the top 10 models, in terms of EF, is as follows: 4 ensembles of MLP, 2 ensembles of SVR, 2 ensembles of ANFIS, and 2 heterogeneous ensembles. None of the individual models was among the top 10 models.

4.4.5. Results Based on Desharnais Dataset. Among the individual models, the SVR model achieved the best performance in terms of MMRE, PRED(25), and EF. By comparing the performance of the homogeneous ensembles of MLP, it can be observed that HM-MLP-[FIS-SC], HM-MLP-[ANFIS-FCM], and HM-MLP-[ANFIS-SC] were the best models based on the EF metric. Moreover, all homogeneous ensembles of MLP except HM-MLP-[MLP] improved the performance of the individual MLP model in terms of EF. By comparing the performance of the homogeneous ensembles of SVR, it can be noticed that HM-SVR-[Avg] and HM-SVR-[WtAvg] were the best. Furthermore, all other homogeneous

FIGURE 13: Histograms of models' EF metric using COCOMO dataset.

FIGURE 14: MMRE versus PRED(25) by each model using COCOMO dataset.

TABLE 8: Models' performance using COCOMO dataset.

Model type	Model	MMRE	PRED(25)	EF
Individual	MLP	250.77	23.1	0.09
	SVR	803.96	7.7	0.01
	ANFIS	532.33	23.1	0.04
Homogeneous ensemble (MLP)	HM-MLP-[Avg]	68.53	23.1	0.33
	HM-MLP-[WtAvg]	68.62	23.1	0.33
	HM-MLP-[MLP]	71.57	23.1	0.32
	HM-MLP-[SVR]	466.60	0.0	0.00
	HM-MLP-[FIS-FCM]	432.81	23.1	0.05
	HM-MLP-[FIS-SC]	102.49	0.0	0.00
	HM-MLP-[ANFIS-FCM]	98.98	15.4	0.15
	HM-MLP-[ANFIS-SC]	307.08	15.4	0.05
Homogeneous ensemble (SVR)	HM-SVR-[Avg]	131.79	7.7	0.06
	HM-SVR-[WtAvg]	131.34	7.7	0.06
	HM-SVR-[MLP]	71.57	23.1	0.32
	HM-SVR-[SVR]	515.51	0.0	0.00
	HM-SVR-[FIS-FCM]	1471.43	0.0	0.00
	HM-SVR-[FIS-SC]	102.49	0.0	0.00
	HM-SVR-[ANFIS-FCM]	305.68	23.1	0.08
	HM-SVR-[ANFIS-SC]	131.46	23.1	0.17
Homogeneous ensemble (ANFIS)	HM-ANFIS-[Avg]	165.57	15.4	0.09
	HM-ANFIS-[WtAvg]	165.57	15.4	0.09
	HM-ANFIS-[MLP]	1059.11	7.7	0.01
	HM-ANFIS-[SVR]	480.86	7.7	0.02
	HM-ANFIS-[FIS-FCM]	2804.98	0.0	0.00
	HM-ANFIS-[FIS-SC]	147249731.42	0.0	0.00
	HM-ANFIS-[ANFIS-FCM]	587.09	15.4	0.03
	HM-ANFIS-[ANFIS-SC]	177.90	30.8	0.17
Heterogeneous ensemble (MLP, SVR, ANFIS)	HT-(MLP, SVR, ANFIS)-[Avg]	109.79	7.7	0.07
	HT-(MLP, SVR, ANFIS)-[WtAvg]	83.81	15.4	0.18
	HT-(MLP, SVR, ANFIS)-[MLP]	881.83	0.0	0.00
	HT-(MLP, SVR, ANFIS)-[SVR]	479.36	7.7	0.02
	HT-(MLP, SVR, ANFIS)-[FIS-FCM]	2359.23	0.0	0.00
	HT-(MLP, SVR, ANFIS)-[FIS-SC]	478644.93	7.7	0.00
	HT-(MLP, SVR, ANFIS)-[ANFIS-FCM]	468.18	30.8	0.07
	HT-(MLP, SVR, ANFIS)-[ANFIS-SC]	135.16	30.8	0.23

ensembles of SVR did not improve the performance of the individual SVR model in terms of EF; they performed worse than it. By comparing the performance of the homogeneous ensembles of ANFIS, it can be observed that HM-ANFIS-[SVR] was the best among them in terms of MMRE, PRED(25), and EF. In addition, only two homogeneous ensembles of ANFIS (i.e., HM-ANFIS-[MLP] and HM-ANFIS-[SVR]) achieved better EF than the individual ANFIS model, whereas the other ensembles of ANFIS were worse than it.

Among the heterogeneous ensemble models, HT-(MLP, SVR, ANFIS)-[Avg] and HT-(MLP, SVR, ANFIS)-[ANFIS-FCM] achieved relatively better performance than the other

heterogeneous ensembles. The distribution of the top 10 models, in terms of EF, is as follows: 1 individual model (SVR), 3 ensembles of MLP, 2 ensembles of SVR, and 4 heterogeneous ensembles. None of the ensembles of ANFIS was among the top 10 models.

4.4.6. Overall Results Based on All Datasets. Ranking of models' performance, in each dataset, on the EF metric, is provided in Table 10, where top 10 models in each dataset are highlighted. Based on this table, we provide the following observations across the five datasets.

FIGURE 15: Histograms of models' EF metric using Desharnais dataset.

FIGURE 16: MMRE versus PRED(25) by each model using Desharnais dataset.

TABLE 9: Models' performance using Desharnais dataset.

Model type	Model	MMRE	PRED(25)	EF
Individual	MLP	78.24	20.0	0.25
	SVR	73.83	33.3	0.45
	ANFIS	105.70	20.0	0.19
Homogeneous ensemble (MLP)	HM-MLP-[Avg]	68.68	26.7	0.38
	HM-MLP-[WtAvg]	69.34	26.7	0.38
	HM-MLP-[MLP]	81.79	20.0	0.24
	HM-MLP-[SVR]	90.79	26.7	0.29
	HM-MLP-[FIS-FCM]	78.94	26.7	0.33
	HM-MLP-[FIS-SC]	62.21	26.7	0.42
	HM-MLP-[ANFIS-FCM]	62.84	26.7	0.42
	HM-MLP-[ANFIS-SC]	62.05	26.7	0.42
Homogeneous ensemble (SVR)	HM-SVR-[Avg]	59.01	40.0	0.67
	HM-SVR-[WtAvg]	59.03	40.0	0.67
	HM-SVR-[MLP]	83.97	20.0	0.24
	HM-SVR-[SVR]	93.03	26.7	0.28
	HM-SVR-[FIS-FCM]	97.45	13.3	0.14
	HM-SVR-[FIS-SC]	82.08	20.0	0.24
	HM-SVR-[ANFIS-FCM]	81.97	20.0	0.24
	HM-SVR-[ANFIS-SC]	76.57	26.7	0.34
Homogeneous ensemble (ANFIS)	HM-ANFIS-[Avg]	115.32	20.0	0.17
	HM-ANFIS-[WtAvg]	105.88	20.0	0.19
	HM-ANFIS-[MLP]	107.53	26.7	0.25
	HM-ANFIS-[SVR]	90.80	26.7	0.29
	HM-ANFIS-[FIS-FCM]	104.29	13.3	0.13
	HM-ANFIS-[FIS-SC]	92.21	6.7	0.07
	HM-ANFIS-[ANFIS-FCM]	4850.60	13.3	0.00
	HM-ANFIS-[ANFIS-SC]	91.54	13.3	0.14
Heterogeneous ensemble (MLP, SVR, ANFIS)	HT-(MLP, SVR, ANFIS)-[Avg]	66.13	33.3	0.50
	HT-(MLP, SVR, ANFIS)-[WtAvg]	66.86	26.7	0.39
	HT-(MLP, SVR, ANFIS)-[MLP]	79.30	33.3	0.42
	HT-(MLP, SVR, ANFIS)-[SVR]	92.49	26.7	0.29
	HT-(MLP, SVR, ANFIS)-[FIS-FCM]	85.29	20.0	0.23
	HT-(MLP, SVR, ANFIS)-[FIS-SC]	63.60	20.0	0.31
	HT-(MLP, SVR, ANFIS)-[ANFIS-FCM]	76.14	40.0	0.52
	HT-(MLP, SVR, ANFIS)-[ANFIS-SC]	68.20	20.0	0.29

(i) None of the individual models was among the top 10 models in three datasets (Miyazaki, Maxwell, and COCOMO). In Albrecht dataset, the MLP model was ranked 6th, and, in Desharnais dataset, the SVR model was ranked 5th.

(ii) HM-MLP-[ANFIS-FCM] and HM-SVR-[ANFIS-SC] models occurred most frequently among the top 10 models across the five datasets. They were among the top 10 models in four out of the five datasets.

(iii) Four models were among the top 10 models in three out of the five datasets. These models are

HM-MLP-[Avg], HM-MLP-[FIS-SC], HM-ANFIS-[WtAvg], and HT-(MLP, SVR, ANFIS)-[WtAvg].

(iv) Nine models were never among the top 10 models in any dataset. These models are ANFIS, HM-MLP-[SVR], HM-MLP-[FIS-FCM], HM-SVR-[FIS-FCM], HM-ANFIS-[SVR], HM-ANFIS-[FIS-FCM], HM-ANFIS-[FIS-SC], HT-(MLP, SVR, ANFIS)-[SVR], and HT-(MLP, SVR, ANFIS)-[FIS-FCM].

(v) At least one of the homogeneous ensembles of MLP was among the top 10 models in each dataset. In addition, at least two of the homogeneous ensembles of SVR were among the top 10 models in each dataset.

TABLE 10: Ranking of models based on EF metric (top 10 models are highlighted).

Model type	Model	Albrecht dataset	Miyazaki dataset	Maxwell dataset	COCOMO dataset	Desharnais dataset
Individual	MLP	6	20	11	12	21
	SVR	26	24	13	24	5
	ANFIS	25	12	27	20	28
Homogeneous ensemble (MLP)	HM-MLP-[Avg]	7	32	2	1	11
	HM-MLP-[WtAvg]	4	31	18	2	12
	HM-MLP-[MLP]	5	16	29	4	23
	HM-MLP-[SVR]	11	23	24	27	16
	HM-MLP-[FIS-FCM]	15	35	19	18	14
	HM-MLP-[FIS-SC]	3	13	8	27	7
	HM-MLP-[ANFIS-FCM]	9	8	31	9	8
	HM-MLP-[ANFIS-SC]	16	33	1	19	6
Homogeneous ensemble (SVR)	HM-SVR-[Avg]	19	10	14	17	1
	HM-SVR-[WtAvg]	20	11	20	16	2
	HM-SVR-[MLP]	21	17	6	3	26
	HM-SVR-[SVR]	10	26	22	27	20
	HM-SVR-[FIS-FCM]	24	14	21	27	32
	HM-SVR-[FIS-SC]	29	2	15	27	25
	HM-SVR-[ANFIS-FCM]	2	22	34	13	24
	HM-SVR-[ANFIS-SC]	1	1	10	7	13
Homogeneous ensemble (ANFIS)	HM-ANFIS-[Avg]	29	4	7	11	30
	HM-ANFIS-[WtAvg]	22	5	9	10	29
	HM-ANFIS-[MLP]	8	21	26	25	22
	HM-ANFIS-[SVR]	12	19	23	23	17
	HM-ANFIS-[FIS-FCM]	17	28	30	27	33
	HM-ANFIS-[FIS-SC]	29	30	12	27	34
	HM-ANFIS-[ANFIS-FCM]	18	9	33	21	35
	HM-ANFIS-[ANFIS-SC]	29	34	4	8	31
Heterogeneous ensemble (MLP, SVR, ANFIS)	HT-(MLP, SVR, ANFIS)-[Avg]	29	6	16	14	4
	HT-(MLP, SVR, ANFIS)-[WtAvg]	23	3	17	6	10
	HT-(MLP, SVR, ANFIS)-[MLP]	28	7	28	27	9
	HT-(MLP, SVR, ANFIS)-[SVR]	27	25	25	22	19
	HT-(MLP, SVR, ANFIS)-[FIS-FCM]	29	27	34	27	27
	HT-(MLP, SVR, ANFIS)-[FIS-SC]	29	18	5	26	15
	HT-(MLP, SVR, ANFIS)-[ANFIS-FCM]	14	29	32	15	3
	HT-(MLP, SVR, ANFIS)-[ANFIS-SC]	13	15	3	5	18

(vi) None of the homogeneous ensembles of ANFIS was among the top 10 models in Desharnais dataset. However, at least one of the homogeneous ensembles of ANFIS was among the top 10 models in the other datasets.

(vii) None of the heterogeneous ensembles was among the top 10 models in Albrecht dataset. However, at least two of the heterogeneous ensembles were among the top 10 models in the other datasets.

(viii) All ensembles models with the nonlinear combiner [FIS-FCM] did not perform well as they were not among the top 10 models in any dataset.

(ix) All ensembles models with the nonlinear combiner [SVR], except HM-SVR-[SVR] model, were not among the top 10 models in any dataset. In case of the HM-SVR-[SVR] model, it was ranked 10th in Albrecht dataset and was not among the top 10 models in the other four datasets.

- (x) By considering the average rank of each model across the datasets, the best five models are (1) HM-SVR-[ANFIS-SC], (2) HM-MLP-[Avg], (3) HT-(MLP, SVR, ANFIS)-[ANFIS-SC], (4) HM-MLP-[FIS-SC], and (5) HT-(MLP, SVR, ANFIS)-[WtAvg].

5. Conclusion

The paper evaluated and compared different homogeneous and heterogeneous ensembles of some optimized hybrid of computational intelligence models for software development effort estimation. Different linear and nonlinear combiners have been used. We have conducted an empirical study to evaluate and compare the performance of these ensembles using five popular datasets. The main findings of this study are as follows. The results confirm that individual models are not reliable as their performance is inconsistent and unstable across different datasets. Although none of the ensemble models was consistently the best, many of them were frequently among the best models for each dataset. The homogeneous ensemble of SVR, with the nonlinear combiner ANFIS-SC, was the best model when considering the average rank of each model across the five datasets. Based on the empirical results, we do not recommend using the following models as they performed worse than the other models across the five datasets: ANFIS, HM-MLP-[SVR], HM-MLP-[FIS-FCM], HM-SVR-[FIS-FCM], HM-ANFIS-[SVR], HM-ANFIS-[FIS-FCM], HM-ANFIS-[FIS-SC], HT-(MLP, SVR, ANFIS)-[SVR], and HT-(MLP, SVR, ANFIS)-[FIS-FCM]. Moreover, we do not recommend using the nonlinear combiner FIS-FCM.

This paper has contributed interesting empirical based insights into the application of different homogeneous and heterogeneous ensemble models in estimating software development effort. Future works include assessment of ensembles of different computational intelligence models and with different combination rules. Homogeneous ensembles of models other than MLP, SVR, and ANFIS can be developed and evaluated. In addition, heterogeneous ensembles with different base models can be also explored and evaluated. In these ensembles, various linear and nonlinear combination rules can be evaluated. An interesting future research is to investigate what might be a best set of base models for heterogeneous ensembles. Another direction of future work is to apply ensemble models in other software engineering prediction problems. This includes classification and regression problems of fault and changeability prediction.

Acknowledgments

The authors would like to acknowledge the support provided by the Deanship of Scientific Research at King Fahd University of Petroleum & Minerals (KFUPM) under Research Grant FT111007. They would like to thank the anonymous reviewers for their insightful comments and suggestions and also Mr. Yasser Khan and Dr. Mohamed El-Attar for their feedback.

References

- [1] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Information and Software Technology*, vol. 54, no. 1, pp. 41–59, 2012.
- [2] R. N. Charette, "Why software fails," *IEEE Spectrum*, vol. 42, no. 9, pp. 36–43, 2005.
- [3] C. López-Martín, C. Yáñez-Márquez, and A. Gutiérrez-Tornés, "Predictive accuracy comparison of fuzzy models for software development effort of small programs," *Journal of Systems and Software*, vol. 81, no. 6, pp. 949–960, 2008.
- [4] I. F. de Barcelos Tronto, J. D. S. da Silva, and N. Sant'Anna, "An investigation of artificial neural networks based prediction systems in software project management," *Journal of Systems and Software*, vol. 81, no. 3, pp. 356–367, 2008.
- [5] S. Grimstad and M. Jørgensen, "Inconsistency of expert judgment-based estimates of software development effort," *Journal of Systems and Software*, vol. 80, no. 11, pp. 1770–1777, 2007.
- [6] M. Jørgensen, "A review of studies on expert estimation of software development effort," *Journal of Systems and Software*, vol. 70, no. 1-2, pp. 37–60, 2004.
- [7] B. Boehm, *Software Engineering Economics*, Prentice-Hall, New Jersey, NJ, USA, 1981.
- [8] A. J. Albrecht and J. E. Gaffney Jr., "Software function, source lines of code and development effort prediction: a software science validation," *IEEE Transactions on Software Engineering*, vol. 9, no. 6, pp. 639–648, 1983.
- [9] L. H. Putnam, "A general empirical solution to the macro sizing and estimating problem," *IEEE Transactions on Software Engineering*, vol. 4, no. 4, pp. 345–361, 1978.
- [10] A. Idri, A. Abran, and T. Khoshgoftaar, "Estimating software project effort by analogy based on linguistic values," in *Proceedings of the 8th IEEE Symposium on Software Metrics*, pp. 21–30, 2002.
- [11] A. Heiat, "Comparison of artificial neural network and regression models for estimating software development effort," *Information and Software Technology*, vol. 44, no. 15, pp. 911–922, 2002.
- [12] E. S. Jun and J. K. Lee, "Quasi-optimal case-selective neural network model for software effort estimation," *Expert Systems with Applications*, vol. 21, no. 1, pp. 1–14, 2001.
- [13] 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, 2000.
- [14] P. C. Pendharkar, G. H. Subramanian, and J. A. Rodger, "A probabilistic model for predicting software development effort," *IEEE Transactions on Software Engineering*, vol. 31, no. 7, pp. 615–624, 2005.
- [15] B. Başkeleş, B. Turhan, and A. Bener, "Software effort estimation using machine learning methods," in *Proceedings of the 22nd International Symposium on Computer and Information Sciences (ISCIS '07)*, pp. 209–214, November 2007.
- [16] K. Srinivasan and D. Fisher, "Machine learning approaches to estimating software development effort," *IEEE Transactions on Software Engineering*, vol. 21, no. 2, pp. 126–137, 1995.
- [17] N.-H. Chiu and S.-J. Huang, "The adjusted analogy-based software effort estimation based on similarity distances," *Journal of Systems and Software*, vol. 80, no. 4, pp. 628–640, 2007.

- [18] M. Shepperd and C. Schofield, "Estimating software project effort using analogies," *IEEE Transactions on Software Engineering*, vol. 23, no. 12, pp. 736–743, 1997.
- [19] 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, 2001.
- [20] K. K. Shukla, "Neuro-genetic prediction of software development effort," *Information and Software Technology*, vol. 42, no. 10, pp. 701–713, 2000.
- [21] A. L. I. Oliveira, "Estimation of software project effort with support vector regression," *Neurocomputing*, vol. 69, no. 13-15, pp. 1749–1753, 2006.
- [22] 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 *Proceedings of the International Joint Conference on Neural Networks (IJCNN '07)*, pp. 1595–1600, August 2007.
- [23] Y. Kultur, B. Turhan, and A. B. Bener, "ENNA: software effort estimation using ensemble of neural networks with associative memory," in *Proceedings of the 16th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (SIGSOFT '08/FSE-16)*, pp. 330–338, November 2008.
- [24] Y. Kultur, B. Turhan, and A. Bener, "Ensemble of neural networks with associative memory (ENNA) for estimating software development costs," *Knowledge-Based Systems*, vol. 22, no. 6, pp. 395–402, 2009.
- [25] L. Minku and X. Yao, "Ensembles and locality: insight on improving software effort estimation," *Information and Software Technology*, vol. 55, no. 8, pp. 1512–1528, 2013, 095058 .
- [26] L. L. Minku and X. Yao, "A principled evaluation of ensembles of learning machines for software effort estimation," in *Proceedings of the 7th International Conference on Predictive Models in Software Engineering (PROMISE '11)*, September 2011.
- [27] E. Kocaguneli, Y. Kultur, and A. Bene, "Combining multiple learners induced on multiple datasets for software effort prediction," in *Proceedings of the International Symposium on Software Reliability Engineering (ISSRE '09)*, 2009.
- [28] 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, 2012.
- [29] M. Elish, "Assessment of voting ensemble for estimating software development effort," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '13)*, pp. 322–327, 2013.
- [30] T. Helmy and A. Fatai, "Hybrid computational intelligence models for porosity and permeability prediction of petroleum reservoirs," *International Journal of Computational Intelligence and Applications*, vol. 9, no. 4, pp. 313–337, 2010.
- [31] R. Harris, X. Yao, G. Brown, and J. Wyatt, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.
- [32] N. Ueda and R. Nakano, "Generalization error of ensemble estimators," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '96)*, pp. 90–95, June 1996.
- [33] Y. Zhao, J. Gao, and X. Yang, "A survey of neural network ensembles," in *Proceedings of the International Conference on Neural Networks and Brain Proceedings (ICNNB '05)*, pp. 438–442, October 2005.
- [34] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust regression for developing software estimation models," *The Journal of Systems and Software*, vol. 27, no. 1, pp. 3–16, 1994.
- [35] K. Maxwell, *Applied Statistics For Software Managers*, Prentice-Hall, New Jersey, NJ, USA, 2002.
- [36] J. Desharnais, *Analyse statistique de la productivité des projets informatiques à partir de la technique des point des fonction [M.S. thesis]*, University of Montreal, 1989.
- [37] R. D. A. Araújo, A. L. I. de Oliveira, and S. C. B. Soares, "A morphological-rank-linear approach for software development cost estimation," in *Proceedings of the 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI '09)*, pp. 630–636, November 2009.