International Journal of Software Innovation
Volume 5 • Issue 4 • October-December 2017

Machine Learning Classification to Effort Estimation for Embedded Software Development Projects

Kazunori Iwata, Department of Business Administration, Aichi University, Nagoya, Japan

Toyoshiro Nakashima, Department of Culture-Information Studies, Sugiyama Jogakuen University, Nagoya, Japan

Yoshiyuki Anan, Process Innovation H.Q, Omron Software Co., Ltd., Kyoto, Japan

Naohiro Ishii, Department of Information Science, Aichi Institute of Technology, Nagoya, Japan

ABSTRACT

This paper discusses the effect of classification in estimating the amount of effort (in man-days) associated with code development. Estimating the effort requirements for new software projects is especially important. As outliers are harmful to the estimation, they are excluded from many estimation models. However, such outliers can be identified in practice once the projects are completed, and so they should not be excluded during the creation of models and when estimating the required effort. This paper presents classifications for embedded software development projects using an artificial neural network (ANN) and a support vector machine. After defining the classifications, effort estimation models are created for each class using linear regression, an ANN, and a form of support vector regression. Evaluation experiments are carried out to compare the estimation accuracy of the model both with and without the classifications using 10-fold cross-validation. In addition, the Games-Howell test with one-way analysis of variance is performed to consider statistically significant evidence.

KEYWORDS

Artificial Neural Network, Classification, Embedded Software, Software Development Process Improvement, Support Vector Regression

INTRODUCTION

The growth and expansion of our information-based society has resulted in an increasing number of information products. In addition, the functionality of these products is becoming ever more complex (Hirayama, 2004; Takagi 2003). Guaranteeing the quality of software is particularly important, because this relates to reliability. It is, therefore, increasingly important for corporations that develop embedded software to realize efficient methods while guaranteeing timely delivery, high quality, and low development costs (Boehm, 1976; Komiyama, 2003; Nakashima, 2004; Ogasawara & Kojima, 2003; Takagi 2003; Tamaru, 2004; Ubayashi, 2004; Watanabe, 2004). Companies and divisions involved in the development of such software are focusing on a variety of improvements, particularly in their processes. Estimating the amount of effort (man-days cost) required for new software projects and guaranteeing product quality are especially important, because the amount of effort is directly related to cost and the product quality affects the reputation of the corporation. In the field of embedded software, there have been various studies on development techniques, management techniques, tools,

DOI: 10.4018/IJSI.2017100102

testing techniques, reuse techniques, real-time operating systems, and other elements. However, there has been little research on the relationships among the scale of the development, amount of effort, and number of errors based on data accumulated from past projects. Previously, we investigated the estimation of total effort and errors using an artificial neural network (ANN), and showed that ANN models are superior to regression analysis models for estimating effort and errors in new projects. We proposed a method to estimate intervals for the number of errors using a support vector machine (SVM) and an ANN. These models were created with data that excluded outliers. The outliers can be identified in practice once the projects have been completed, and so they should not be excluded during the creation of models and when estimating the effort required. This paper presents classifications for embedded software development projects based on whether the amount of effort is an outlier or not using an ANN and an SVM. After the classification stage, we establish effort estimation models for each class using linear regression (LR), an ANN, and ε-support vector regression (ε -SVR). Evaluation experiments are carried out to compare the estimation accuracy of the model both with and without classification using 10-fold cross-validation and by applying the Games-Howell test with one-way analysis of variance (ANOVA).

The rest of this paper is organized as follows. First, we explain the related works. Second, we show datasets used in this paper. After that, we place our work and evaluation experiments. As a result, we conclude the paper.

RELATED WORKS

Support Vector Regression

SVR uses the same principles as SVM for classification, albeit with a few minor differences. The ε -SVR (Alex & Bernhard, 2004) regression method uses an ε -insensitive loss function to solve regression problems. This approach attempts to find a continuous function in which as many data points as possible lie within the ϵ -wide insensitivity tube. ε -SVR is used to estimate the amount of effort required for software projects (Oliveira, 2006). This approach has been tested using the well-known NASA software project dataset (John & Victor, 1981; Shin & Goel, 2000). However, these studies did not investigate the parameters of ε -SVR. The effectiveness of the SVM (and SVR) using the resulting continuous function depends on the kernel parameter (^{3}) and soft margin parameter (*C*) (Cortes & Vapnik, 1995). In addition, the value of ϵ affects the estimations given by ε -SVR.

We proposed a three-dimensional grid search to find the most appropriate combination of these parameters (Iwata, Liebman, Stone, Nakashima, Anan & Ishii, 2015). Our method improved the mean magnitude of relative error (*MMRE*, see Equation (3) in the section "Evaluation Criteria") from 0.165 (Cortes & Vapnik, 1995) to 0.149 using leave-one-out cross-validation (Shin & Goel, 2000).

Artificial Neural Networks

In earlier papers, we showed that ANN models are superior to regression analysis models for estimating the effort and errors in new projects (Iwata, Nakashima, Anan & Ishii, 2008). In addition, we proposed a method for reducing the margin of error (Iwata, Nakashima, Anan & Ishii, 2010; Iwata, Liebman, Stone, Nakashima, Anan & Ishii, 2015). However, outliers are excluded during the creation of the models, because they may be detrimental to performance. These outliers can be identified in practice once the projects have been completed, and so they should not be excluded from the model creation process or when estimating the amount of effort.

Our Contribution

The above algorithms have a certain level of estimation accuracy for data in which outliers are excluded. The outliers negatively affect the estimation, but cannot be detected before the projects have been completed. Therefore, in this paper, we propose a two-step method for reducing the estimation errors using an ANN and an SVM. Projects are classified according to whether the amount of effort is an outlier, with attributes that can be measured before the projects start used for the classification. After the classification stage, we establish effort estimation models for each class using LR, an ANN, and ε -SVR. We have studied the methods and states in Iwata, Nakashima, Anan & Ishii (2016). This paper is the extended version of the paper.

DATASETS AND OUTLIERS

Original Datasets

Using the following data from a large software company, we created models to estimate the amount of planning effort (*Eff*).

Eff: "The amount of effort," which indicates the cost, in man-days, of the review process for software development projects.
V_{new}: "Volume of newly added," which denotes the number of steps in the newly generated functions of the target project.
V_{modify}: "Volume of modification," which denotes the number of steps modified or added to existing functions to use the target project.
V_{survey}: "Volume of original project," which denotes the original number of steps in the modified functions and the number of steps deleted from the functions.
V_{reuse}: "Volume of reuse," which denotes the number of steps in a function that confirm an external method and are applied to the target project design without confirming the internal content.

Detection of Outliers

This paper examines the classification of outliers in terms of the amount of effort required in a project. Figure 1 shows the distribution of the amount of effort, and Figure 2 is a boxplot of this metric. The lowest datum of the boxplot is within 1.5 times the interquartile range (IQR) of the lower quartile, and the highest datum is within 1.5 IQR of the upper quartile. The outliers are denoted by circles. The Y coordinate values of the outliers have no meaning. The varying the values is used to improve understanding the distribution of the outliers. Of the total of 1416 data points, 146 are outliers. Each value of the boxplot is shown in Table 1.

EFFORT ESTIMATION MODELS

The effort estimation models have two steps-classification and estimation

Model Creation

The data are divided into two groups of normal values and outliers on the basis of the amount of effort. The classification methods are created using an ANN and an SVM. The estimation methods are established for both normal values and outliers using LR, the ANN, and ε -SVR. Figure 3 illustrates the process of model creation.

Effort Estimation

As the first step in estimating the amount of effort in a project, the data are classified using the classification methods. The step divides the date between candidates for normal values and for outliers. The amount of effort is then estimated using the method corresponding to the classification result (Figure 4). The volumes described in the section "Original Datasets" are used to classify a project and to estimate the amount of effort.

Figure 1. Distribution of the Amount of Effort in Intervals 500

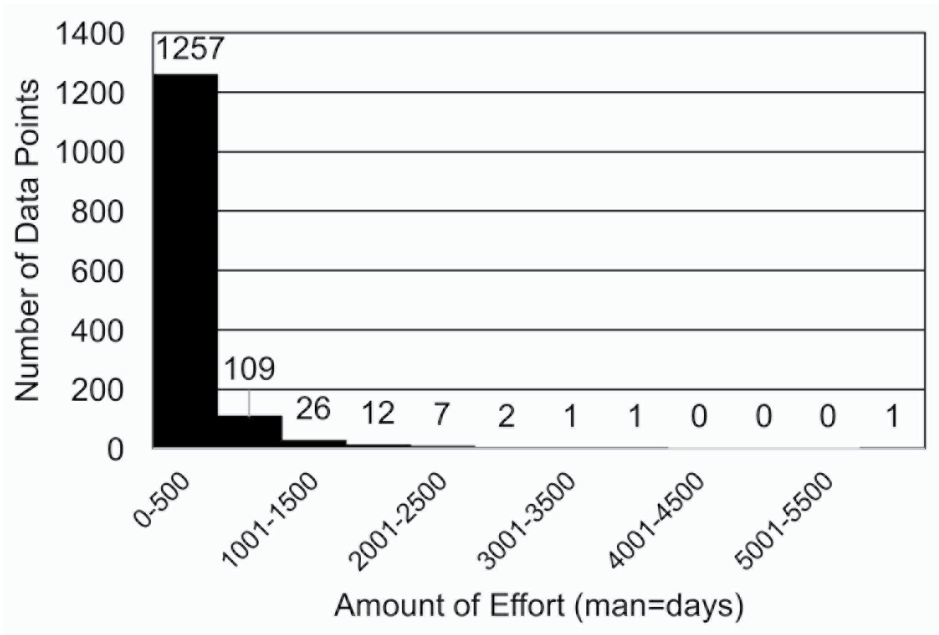


Figure 2. Boxplot of the Amount of Effort

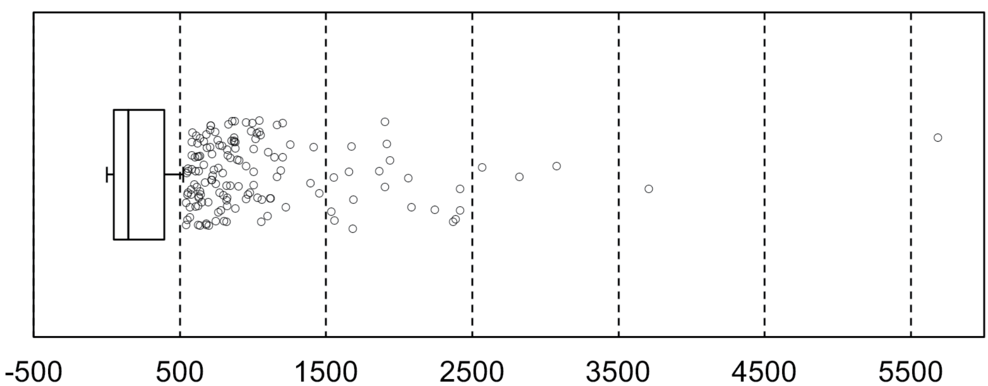


Table 1. Detailed Information of the Boxplot in Figure 1

	Value
IQR	197.070
Minimum	0.000
Lower quartile	49.430
Median	98.500
Upper quartile	246.500
Maximum	542.105

Figure 3. Model Creation

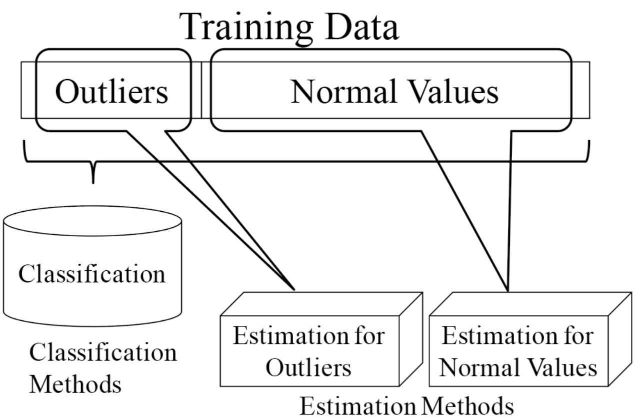
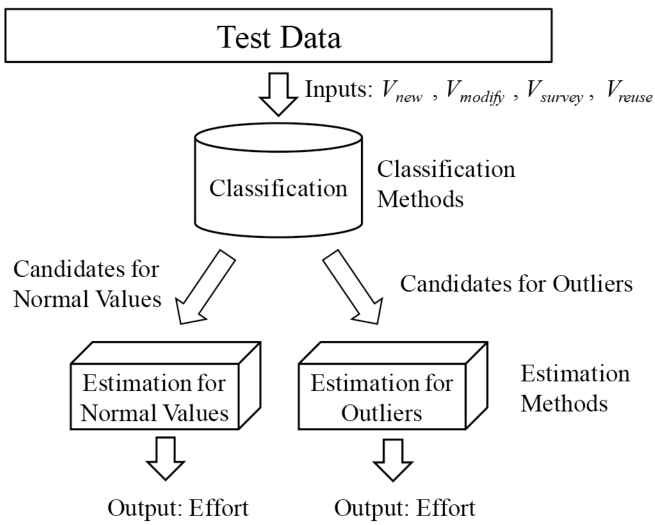


Figure 4. Effort Estimation



EVALUATION EXPERIMENT

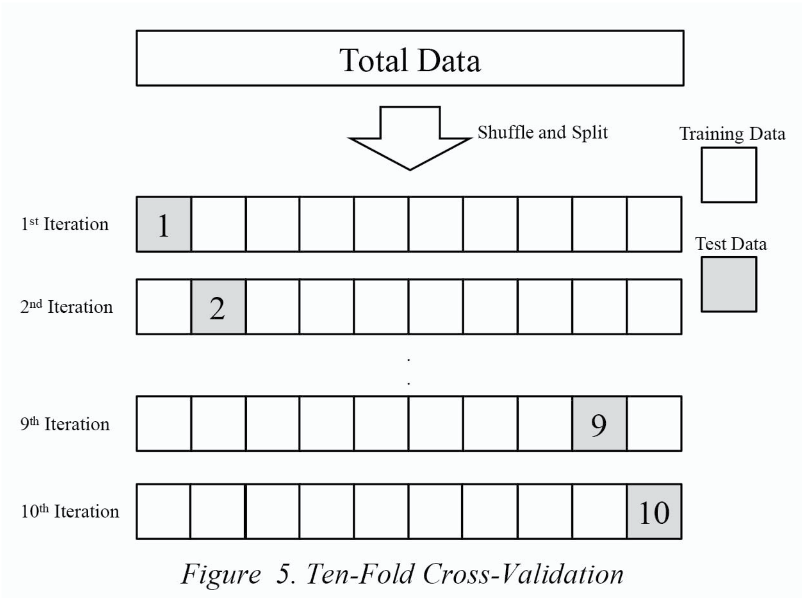
Data Used in Evaluation Experiment

To evaluate the performance of the proposed technique, we performed 10-fold cross-validation on data from 1416 real projects. The original data were randomly partitioned into 10 equally sized subsamples (with each subsample having data from 141 or 142 projects). One of the subsamples was used as the validation data for testing the model, while the remaining nine subsamples were used as training data. The cross-validation process was repeated ten times, with each of the ten subsamples used exactly once as validation data. An example of 10-fold cross-validation is shown in Figure 5.

Classification Accuracy

The accuracy of the classification is affected by the estimation model used. The results of classification by ANN and SVM are shown in Tables 2 are 3, respectively. The values in the table represent the

Figure 5. Ten-Fold Cross-Validation



aggregate over 10 experiments. The diagonal values in the tables indicate the successfully classified projects. ANN classification achieved a 94.35% (= (1236 + 100) / 1416) success rate, and SVM classification attained a 95.34% (= (1247 + 103) / 1416) success rate. These results illustrate that the SVM gives slightly better results, but the difference is not significant. Therefore, both methods can be used to classify the projects.

Models for Experiments

We compared the accuracy of the following models in estimating the amount of effort required in the projects. Models 1) - 3) do not use the proposed classification methods, and estimate the amount of effort involved in a project directly. In contrast, models 4) - 6) divide the projects into two groups according to the ANN classification results. Models 7) - 9) divide the projects into two groups according to the SVM classification results.

Model 1) LR model without classification (LR w/o CL)
Model 2) ANN model without classification (ANN w/o CL)
Model 3) ε -SVR model without classification (ε -SVR w/o CL)
Model 4) LR model with ANN classification (LR w/ ANN)

Table 2. Classification Results by ANN

		Expected Classes		
		Normal Values	Outliers	Total
True Classes	Normal Values	1236	34	1270
	Outliers	46	100	146
	Total	1282	134	1416

Table 3. Classification Results by SVM

		Expected Classes		
		Normal Values	**Outliers**	**Total**
True Classes	Normal Values	1247	23	1270
	Outliers	43	103	146
	Total	1290	126	1416

Model 5) ANN model with ANN classification (ANN w/ ANN)
Model 6) ε -SVR model with ANN classification (ε -SVR w/ ANN)
Model 7) LR model with SVM classification (LR w/ SVM)
Model 8) ANN model with SVM classification (ANN w/ SVM)
Model 9) ε -SVR model with SVM classification (ε -SVR w/ SVM)

Evaluation Criteria

The following six criteria were used as performance measures for the effort estimation models (Shin & Goel, 2000). *PRED(p)* is the prediction level in Equation (5) (Conte, Dunsmore & Shen, 1986). For the measures 1) and 3), smaller values of the evaluation criterion indicate higher relative accuracy. In contrast, a larger value of *MPRED(25)* implies higher relative accuracy. The value of $\frac{X-\hat{X}}{X}$ is regarded as 1 if *X* is equal to 0 in the calculation of *MMRE* and *SDRE*. The actual value is expressed as X, while the predicted value is expressed as $\hat{X}$.

1. Mean of absolute errors (*MAE*).
2. Standard deviation of absolute errors (*SDAE*).
3. Mean magnitude of relative errors (*MMRE*).
4. Standard deviation of relative errors (*SDRE*).
5. *MPRED(25)* is the mean of *PRED(25)*.

SDPRED(25) is the standard deviation of *PRED(25)*.

$$MAE = \frac{1}{n}\sum\left|X-\hat{X}\right| \tag{1}$$

$$SDRE = \sqrt{\frac{1}{n-1}\sum\left(\left|X-\hat{X}\right|-MAE\right)^2} \tag{2}$$

$$MMRE = \frac{1}{n}\sum\left|\frac{X-\hat{X}}{X}\right| \tag{3}$$

$$SDRE = \sqrt{\frac{1}{n-1}\sum\left(\left|\frac{X-\hat{X}}{X}\right| - MMRE\right)^2} \tag{4}$$

$$PRED\left(q\right) = \frac{k}{n} \tag{5}$$

where for the set of *n* estimates, let *k* denote the number of estimates for which the magnitude of relative errors is less than or equal $q\%$.

RESULTS AND DISCUSSION

For each model, the experimental results for all projects using 10-fold cross-validation are presented in Tables 4, 5 and 6.

Tables 7, 8 and 9 give the experimental results for projects considered to have normal values. The projects for models were detected using the classification methods. In contrast, the projects for models 1), 2) and 3) were extracted from the full dataset according to the true classes. Tables 10, 11 and 12 present the experimental results for projects considered to be outliers. These outlier projects were detected in a similar manner to the projects with normal values.

We compared the accuracy of the models using the Games-Howell test. This is a post hoc method for one-way ANOVA with unequal group sizes and unequal variances. The test is based on Welch's *t*-test and uses the studentized range statistic (*q*-distribution). This distribution is similar to the *t*-distribution from the *t*-test. If the *p*-value is less than or equal to the significance level, the null hypothesis is rejected. The null hypothesis in our experiment can be interpreted as "there is no difference between the means of the estimation errors in the pair of models being compared''. The results of the one-way ANOVA prior to the Games-Howell test reveals that the Games-Howell test for outliers *AE* is not warranted and should not be carried out at a significance level of 0.05.

The results of the Games-Howell test are given as underlined values in Tables 4-12. They indicate statistically significant differences between with and without a classification, and illustrate the effect of classification on estimating the amount of effort. In addition, the tests indicate that the ANN w/ ANN, ANN w/ SVM, SVR w/ ANN and SVR w/ SVM can estimate the amount of effort

Table 4. Experimental Results of Absolute Errors (AE) for All Data

Models	# of Projects	*MAE*	*SDAE*	95% Confidence Interval
LR w/o CL	1416	115.795	206.506	[105.030, 126.560]
ANN w/o CL	1416	114.347	239.777	[101.848, 126.847]
SVR w/o CL	1416	129.110	265.647	[115.262, 142.958]
LR w/ ANN	1416	126.488	218.713	[115.086, 137.889]
ANN w/ ANN	1416	111.713	218.951	[100.300, 123.127]
SVR w/ ANN	1416	105.209	222.991	[93.584, 116.833]
LR w/ SVM	1416	123.919	225.216	[112.179, 135.660]
ANN w/ SVM	1416	109.504	236.506	[97.175, 121.833]
SVR w/ SVM	1416	**<u>101.623</u>**	233.828	[89.433, 113.812]

Table 5. Experimental Results of Relative Errors (RE) for All Data

Models	# of Projects	*MMRE*	*SDRE*	95% Confidence Interval
LR w/o CL	1416	1.886	5.789	[1.584, 2.188]
ANN w/o CL	1416	1.220	3.010	[1.063, 1.377]
SVR w/o CL	1416	0.948	1.789	[0.855, 1.042]
LR w/ ANN	1416	1.844	5.597	[1.552, 2.136]
ANN w/ ANN	1416	0.958	2.358	[0.835, 1.081]
SVR w/ ANN	1416	0.944	2.331	[0.823, 1.066]
LR w/ SVM	1416	1.834	5.596	[1.542, 2.126]
ANN w/ SVM	1416	0.951	2.361	[0.828, 1.074]
SVR w/ SVM	1416	0.933	2.333	[0.812, 1.055]

Table 6. Experimental Results of MPRED (25) for All Data

Models	# of Projects	*MPRED(25)*	*SDPRED(25)*	95% Confidence Interval
LR w/o CL	1416	0.258	0.037	[0.256, 0.260]
ANN w/o CL	1416	0.293	0.056	[0.290, 0.296]
SVR w/o CL	1416	0.303	0.040	[0.301, 0.305]
LR w/ ANN	1416	0.221	0.047	[0.218, 0.223]
ANN w/ ANN	1416	0.299	0.055	[0.296, 0.302]
SVR w/ ANN	1416	0.331	0.049	[0.329, 0.334]
LR w/ SVM	1416	0.222	0.043	[0.219, 0.224]
ANN w/ SVM	1416	0.302	0.054	[0.299, 0.305]
SVR w/ SVM	1416	0.342	0.042	[0.339, 0.344]

Table 7. Experimental Results of Absolute Errors (AE) for Normal Values

Models	# of Projects	*MAE*	*SDAE*	95% Confidence Interval
LR w/o CL	1270	79.983	110.539	[73.899, 86.069]
ANN w/o CL	1270	76.798	139.113	[69.140, 84.456]
SVR w/o CL	1270	88.400	139.778	[80.705, 96.095]
LR w/ ANN	1282	87.716	119.111	[81.190, 94.243]
ANN w/ ANN	1282	73.462	108.244	[67.531, 79.393]
SVR w/ ANN	1282	71.460	117.921	[64.998, 77.921]
LR w/ SVM	1290	91.605	156.889	[83.036, 100.175]
ANN w/ SVM	1290	77.549	159.690	[68.827, 86.272]
SVR w/ SVM	1290	74.590	166.990	[65.468, 83.711]

Table 8. Experimental Results of Relative Errors (RE) for Normal Values

Models	# of Projects	*MMRE*	*SDRE*	95% Confidence Interval
LR w/o CL	1270	2.057	6.089	[1.722, 2.392]
ANN w/o CL	1270	1.312	3.164	[1.138, 1.486]
SVR w/o CL	1270	1.008	1.877	[0.905, 1.112]
LR w/ ANN	1282	1.921	5.821	[1.602, 2.240]
ANN w/ ANN	1282	**0.996**	2.462	[0.861, 1.131]
SVR w/ ANN	1282	0.954	2.364	[0.824, 1.083]
LR w/ SVM	1290	1.913	5.804	[1.596, 2.230]
ANN w/ SVM	1290	0.992	2.455	[0.858, 1.126]
SVR w/ SVM	1290	0.948	2.358	[0.819, 1.076]

Table 9. Experimental Results of MPRED (25) for Normal Values

Models	# of Projects	*MPRED(25)*	*SDPRED(25)*	95% Confidence Interval
LR w/o CL	1270	0.249	0.039	[0.247, 0.251]
ANN w/o CL	1270	0.286	0.055	[0.283, 0.289]
SVR w/o CL	1270	0.303	0.047	[0.300, 0.306]
LR w/ ANN	1282	0.213	0.029	[0.211, 0.214]
ANN w/ ANN	1282	0.296	0.028	[0.294, 0.297]
SVR w/ ANN	1282	0.327	0.032	[0.325, 0.328]
LR w/ SVM	1290	0.211	0.029	[0.209, 0.212]
ANN w/ SVM	1290	0.295	0.028	[0.293, 0.296]
SVR w/ SVM	1290	0.332	0.026	[0.331, 0.334]

Table 10. Experimental Results of Absolute Errors (AE) for Outliers

Models	# of Projects	*MAE*	*SDAE*	95% Confidence Interval
LR w/o CL	146	427.306	447.521	[354.104, 500.508]
ANN w/o CL	146	440.973	521.440	[355.680, 526.267]
SVR w/o CL	146	483.228	613.902	[382.810, 583.646]
LR w/ ANN	134	497.420	466.666	[417.681, 577.159]
ANN w/ ANN	134	477.673	496.549	[392.828, 562.519]
SVR w/ ANN	134	428.093	526.561	[338.120, 518.066]
LR w/ SVM	126	454.755	444.836	[376.324, 533.186]
ANN w/ SVM	126	436.656	500.036	[354.864, 518.448]
SVR w/ SVM	126	378.389	494.845	[291.141, 465.637]

Table 11. Experimental Results of Relative Errors (RA) for Outliers

Models	# of Projects	*MMRE*	*SDRE*	95% Confidence Interval
LR w/o CL	146	0.391	0.253	[0.349, 0.432]
ANN w/o CL	146	0.408	0.338	[0.353, 0.463]
SVR w/o CL	146	0.417	0.336	[0.362, 0.472]
LR w/ ANN	134	1.106	2.488	[0.681, 1.531]
ANN w/ ANN	134	0.591	0.767	[0.460, 0.722]
SVR w/ ANN	134	0.852	1.987	[0.513, 1.192]
LR w/ SVM	126	1.021	2.500	[0.580, 1.462]
ANN w/ SVM	126	0.528	0.853	[0.389, 0.668]
SVR w/ SVM	126	0.785	2.056	[0.423, 1.148]

Table 12. Experimental Results of MPRED(25) for Outliers

Models	# of Projects	*MPRED(25)*	*SDPRED(25)*	95% Confidence Interval
LR w/o CL	146	0.327	0.082	[0.314, 0.340]
ANN w/o CL	146	0.354	0.092	[0.339, 0.369]
SVR w/o CL	146	0.295	0.081	[0.282, 0.308]
LR w/ ANN	134	0.298	0.126	[0.276, 0.319]
ANN w/ ANN	134	0.334	0.158	[0.307, 0.361]
SVR w/ ANN	134	0.379	0.124	[0.358, 0.400]
LR w/ SVM	126	0.331	0.112	[0.311, 0.351]
ANN w/ SVM	126	0.379	0.159	[0.353, 0.405]
SVR w/ SVM	126	0.439	0.112	[0.419, 0.458]

more accurately than LR. However, Tables 10, 11 and 12 suggest that the classification process does not affect the estimation of the amount of effort in outliers. This is because the number of outliers is so small that there can be no statistically significant differences between model pairs.

CONCLUSION

This paper has examined the effect of classification in estimating the amount of effort required in software development projects. Embedded software development projects have been classified according to whether the amount of effort is an outlier using an ANN and an SVM. After the classification stage, effort estimation models for each class were created using LR, an ANN, and ε -SVR.

Evaluation experiments were conducted to compare the estimation accuracy of the models both with and without the classification step using 10-fold cross-validation. The results show that the classification process has a positive effect. In addition, the Games-Howell test was performed to identify statistically significant evidence. The results indicate statistically significant differences between certain pairs of models.

In future research, we will investigate the following:

- Having implemented a model to estimate the final amount of effort required in new projects, we plan to estimate the partial effort at various stages (e.g., halfway) in the project development process.
- We intend to consider a more complex method, such as a Bayesian network, to improve the accuracy of the proposed technique.
- More data are needed to further support our work. In particular, data for projects including outliers are essential to ameliorate the models.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP16K00310.

REFERENCES

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. N., and Csaki, F. (eds.), *Proceedings of the 2nd International Symposium on Information Theory* (pp. 267-281).

Alex, J. S., & Bernhard, S. (2004). A Tutorial on Support Vector Regression [Kluwer Academic Publishers.]. *Statistics and Computing*, *14*(3), 199–222. doi:10.1023/B:STCO.0000035301.49549.88

Aoki, S. (2007). In testing whether the means of two populations are different (in Japanese), Retrieved from http://aoki2.si.gunma-u.ac.jp/lecture/BF/index.html

Boehm, B. W. (1976). Software engineering. *IEEE Transactions on Software Engineering*, *C-25*(12), 1226–1241.

Conte, S. D., Dunsmore, H. E. & Shen, Y. E. (1986). Software Engineering Metrics and Models: Benjamin-Cummings Publishing Co., Inc.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. doi:10.1007/BF00994018

Hirayama, M. (2004). Current state of embedded software (in Japanese). *Journal of Information Processing Society of Japan*, *45*(7), 677–681.

Iwata, K., Liebman, E., Stone, P., Nakashima, T., Anan, Y., & Ishii, N. (2015). Bin-Based Estimation of the Amount of Effort for Embedded Software Development Projects with Support Vector Machines. In L. Roger (Ed.), *Computer and Information Science 2015, Studies in Computational Intelligence* (Vol. 614, pp. 157–169). Springer International Publishing. doi:10.1007/978-3-319-23467-0_11

Iwata, K., Nakashima, T., Anan, Y., & Ishii, N. (2008). Error Estimation Models Integrating Previous Models and Using Artificial Neural Networks for Embedded Software Development Projects, *In Proceedings of 20th IEEE International Conference on Tools with Artificial Intelligence ICTAI '08* (pp. 371-378). IEEE Press.

Iwata, K., Nakashima, T., Anan, Y., & Ishii, N. (2010). Improving Accuracy of an Artificial Neural Network Model to Predict Effort and Errors in Embedded Software Development Projects. In L. Roger (Ed.), *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2010, Studies in Computational Intelligence* (Vol. 295, pp. 11–21). Springer Berlin Heidelberg. doi:10.1007/978-3-642-13265-0_2

Iwata, K., Nakashima, T., Anan, Y., & Ishii, N. (2011-08), Clustering and Analyzing Embedded Software Development Projects Data using Self-Organizing Maps, In L. Roger (Ed.), Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2011, SCI (Vol. 377, pp. 47-59). Springer Berlin Heidelberg.

Iwata, K., Nakashima, T., Anan, Y., & Ishii, N. (2016), Effort Estimation for Embedded Software Development Projects by Combining Machine Learning with Classification. *Proceedings of 3rd ACIS International Conference on Computational Science/Intelligence and Applied Information CSII '16* (pp. 265-270). IEEE Press. doi:10.1109/ACIT-CSII-BCD.2016.058

Iwata, K., Nakashima, T., Anan, Y., & Ishii, N. (2011-11), Effort Prediction Models Using Self-Organizing Maps for Embedded Software Development Projects. *Proceedings of 23rd IEEE International Conference on Tools with Artificial Intelligence ICTAI '11* (pp. 142-147). IEEE Press.

John, W. B., & Victor, R. B. (1981). A meta-model for software development resource expenditures. *Proceedings of the 5th international conference on Software engineering ICSE '81* (pp. 107-116). IEEE Press, Piscataway, NJ, USA.

Komiyama, T. (2003). Development of foundation for effective and efficient software process improvement (in Japanese). *Journal of Information Processing Society of Japan*, *44*(4), 341–347.

Nakamoto, Y., Takada, H., & Tamaru, K. (1997). Current state and trend in embedded systems (in Japanese). *Journal of Information Processing Society of Japan*, *38*(10), 871–878.

Nakashima, S. (2004). Introduction to model-checking of embedded software (in Japanese). *Journal of Information Processing Society of Japan*, *45*(7), 690–693.

Ogasawara, H., & Kojima, S. (2003). Process improvement activities that put importance on stay power (in Japanese). *Journal of Information Processing Society of Japan*, *44*(4), 334–340.

Oliveira, L. I. A. (2006). Estimation of software project effort with support vector regression. *Neurocomputing*, *69*(13-15), 1749–1753. doi:10.1016/j.neucom.2005.12.119

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3-4), 591–611. doi:10.1093/biomet/52.3-4.591

Shin, M., & Goel, L. A. (2000). Empirical data modeling in software engineering using radial basis functions. IEEE Transactions on Software Engineering, 26(6), 567-567.

Boehm, B. W., Madachy, R., & Steece, B. (2000). Software Cost Estimation with Cocomo ii. NJ: Prentice Hall.

Student. (1908). The probable error of a mean. *Biometrika, 6*(1), 1-25.

Takagi, Y. (2003). A case study of the success factor in large-scale software system development project (in Japanese). *Journal of Information Processing Society of Japan*, *44*(4), 348–356.

Tamaru, K. (2004). Trends in software development platform for embedded systems (in Japanese). *Journal of Information Processing Society of Japan*, *45*(7), 699–703.

Ubayashi, N. (2004). Modeling techniques for designing embedded software (in Japanese). *Journal of Information Processing Society of Japan*, *45*(7), 682–692.

Watanabe, H. (2004). Product line technology for software development (in Japanese). *Journal of Information Processing Society of Japan*, *45*(7), 694–698.

Welch, B. L. (1947). The generalization of student's problem when several different population variances are involved. *Biometrika*, *34*(28). PMID:20287819