# How Effective is Tabu Search to Configure Support Vector Regression for Effort Estimation?

A. Corazza, S. Di Martino
University of Napoli "Federico II"
Via Cintia, 80126
Napoli, Italy
+39 081 679272
{corazza, dimartino}@na.infn.it

F. Ferrucci, C. Gravino, F. Sarro
University of Salerno
Via Ponte don Melillo, 84084
Fisciano (SA), Italy
+39 089 963374
{fferrucci, gravino, fsarro}@unisa.it

E. Mendes
University of Auckland
Private Bag 92019
Auckland, New Zeland
0064 9 3737599 ext. 86137
emilia@cs.auckland.ac.nz

## ABSTRACT

**Background.** Recent studies have shown that Support Vector Regression (SVR) has an interesting potential in the field of effort estimation. However applying SVR requires to carefully set some parameters that heavily affect the prediction accuracy. No general guidelines are available to select these parameters, whose choice also depends on the characteristics of the data set used. This motivates the work described in this paper. **Aims**. We have investigated the use of an optimization technique in combination with SVR to select a suitable subset of parameters to be used for effort estimation. This technique is named Tabu Search (TS), which is a meta-heuristic approach used to address several optimization problems. **Method**. We employed SVR with linear and RBF kernels, and used variables' preprocessing strategies (i.e., logarithmic). As for the data set, we employed the Tukutuku cross-company database, which is widely adopted in Web effort estimation studies, and performed a hold-out validation using two different splits of the data set. As benchmark, results are compared to those obtained with Manual StepWise Regression, Case-Based Reasoning, and Bayesian Networks. **Results**. Our results show that TS provides a good choice of parameters, so that the combination of TS and SVR outperforms any other technique applied on this data set. **Conclusions**. The use of the meta-heuristic Tabu Search allowed us to obtain (I) an automatic choice of the parameters required to run SVR, and (II) a significant improvement on prediction accuracy for SVR. While we are not guaranteed that this is the global optimum, the results we are presenting are the best performance ever obtained on the problem at the hand, up to now. Of course, the experimental results here presented should be assessed on further data. However, they are surely interesting enough to suggest the use of SVR among the techniques that are suitable for effort estimation, especially when using a cross-company database.

## Categories and Subject Descriptors

D.2.9 [**Management**]: Cost estimation, Productivity. D.2.8 [**Metrics**]: Process metrics, Product metrics.


*PROMISE2010*, Sep 12-13, 2010. Timisoara, Romania


## General Terms

Measurement, Experimentation.

## Keywords

Development Effort Estimation, Empirical Studies, Tabu Search, Support Vector Regression, Support Vector Machines.

## 1. INTRODUCTION

Support Vector Regression (SVR) is an approach based on Support Vector Machines, a new generation of Machine Learning algorithms, suitable for predictive data modeling problems [28], [30]. Our previous work showed that the application of SVR using a cross-company data set can be effective for Web effort estimation [8][9]. Indeed, SVR's flexibility, by enabling the use of different kernels and parameter settings, allows for the learning mechanism to suit well the characteristics of different chunks of data, typical of cross-company datasets. As a matter of fact, our previous work showed that a kernel choice and a corresponding combination of parameters enabled SVR to outperform any other prediction technique on that specific cross-company data set [8][9].

However, the flexibility offered by SVR, if not properly addressed, can be interpreted as a drawback of the approach, thus limiting its applicability. For example, an inappropriate choice of parameters setting (usually the "hyper-parameters" $C$ and $\varepsilon$, and the kernel parameter $\gamma$, detailed in the following) can lead to over- or under-fitting, heavily worsening the performance of the method [4][17]. No general guidelines are available to select these parameters [28], [32], [33], mainly because the appropriate setting depends on the characteristics of the employed data set. The problem is further complicated by the fact that there is interaction among parameters, so that a simpler separate optimization of each parameter is not suitable.

The issues abovementioned motivated us to investigate a technique able to jointly optimize all SVR parameters. Note that an extensive exploration of all the possible settings is not computationally affordable, as the search space is too large.

The objective of this paper is therefore to investigate the use of Tabu Search (TS) [15] to identify the most suitable parameters for SVR, to be used for effort estimation. TS is a meta heuristic approach, used to address several optimization problems. In the

empirical study we carried out, we employed SVR with linear and RBF kernels, the two that provided the best results in our previous research [8][9]. Moreover, we used both unprocessed data and a variables' preprocessing strategy (i.e., logarithmic transformation).

As for the data set, we employed the Tukutuku database [22], which is widely adopted in Web effort estimation studies. In particular, we employed the same two random splits of the data set that were used in previous researches. In this way, we were enabled to compare the prediction results among different studies, on the same data [13][20][21][24][26]. The two training sets contained data on 130 projects each, randomly selected from the Tukutuku database, and two validation sets, each containing the remaining 65 projects. As benchmark, other than the previous results obtained with SVR, we employed also Manual StepWise Regression (MSWR) and Case-Based Reasoning (CBR) due to their frequent use in Web & software effort estimation studies, Bayesian Networks as used in [20] and the Mean effort and the Median effort of the training sets.

Consequently, the research questions addressed in this paper are the following:

- Is Tabu Search able to effectively set Support Vector Regression configuration parameters?
- Are the effort predictions obtained by using the combination of Tabu Search and Support Vector Regression significantly superior to the ones obtained by other techniques?

The remainder of the paper is organized as follows. Section 2 describes the data set employed in our empirical study, followed by a description in Section 3 of the SVR approach, together with information on how we set-up some configuration parameters for the evaluations by exploiting TS. Section 4 presents the validation method and evaluation criteria employed to assess the effectiveness of the predictions. Results are discussed in Section 5, followed by our conclusions and comments on future work in Section 6.

## 2. DATA SET DESCRIPTION

In this section the Tukutuku database [22] is described. This database is part of the Tukutuku project, which aims to gather data from completed Web projects, to develop Web cost estimation models and to benchmark productivity across and within Web Companies [22]. The Tukutuku database includes information on Web hypermedia systems and Web applications [4]. The former are characterized by the authoring of information using nodes (chunks of information), links (relations between nodes), anchors, access structures (for navigation) and its delivery over the Web. In addition, typical developers are writers, artists and organizations that wish to publish information on the Web and/or CD-ROMs without the need to use programming languages such as Java. Conversely, the latter represents software applications that depend on the Web or use the Web's infrastructure for execution and are characterized by functionality affecting the state of the underlying business logic. Web applications usually include tools suited to handle persistent data, such as local file system, (remote) databases, or Web Services. Typical developers are young programmers fresh from a Computer Science or Software Engineering degree, managed by more senior staff.

The Tukutuku database has data on 195 projects, where:

- Projects came mostly from 10 different countries, mainly New Zealand (47%), Italy (17%), Spain (16%), Brazil (10%), United States (4%), England (2%), and Canada (2%).
- Project types are new developments (65.6%) or enhancement projects (34.4%).
- About dynamic technologies, PHP is used in 42.6% of the projects, ASP (VBScript or .Net) in 13.8%, Perl in 11.8%, J2EE in 9.2%, while 9.2% of the projects used other solutions.
- The remaining projects used only HTML and/or Javascript.

Each Web project in the database is characterized by process and product variables [22]. Summary statistics for the numerical variables from the Tukutuku database are given in Table 1, while Table 2 summarizes the number and percentages of projects for the categorical variables. Note that those variables are binary, and for our analysis their values have been coded by means of dummy variables, namely 1 (for "new" and "no") and 2 (for "enhancement" and "yes"). We avoided 0 and negative values, that would create problems with logarithmic transformation.

**Table 1. Summary Statistics for numerical variables of Tukutuku database**

| Variable | Mean | Median | Std. Dev | Min | Max |
|---|---|---|---|---|---|
| nlang | 3.9 | 4 | 1.4 | 1 | 8 |
| DevTeam | 2.6 | 2 | 2.4 | 1 | 23 |
| TeamExp | 3.8 | 4 | 2.0 | 1 | 10 |
| TotEff | 468.1 | 88 | 938.5 | 1.1 | 5,000 |
| TotWP | 69.5 | 26 | 185.7 | 1 | 2,000 |
| NewWP | 49.5 | 10 | 179.1 | 0 | 1,980 |
| TotImg | 98.6 | 40 | 218.4 | 0 | 1,820 |
| NewImg | 38.3 | 1 | 125.5 | 0 | 1,000 |
| Fots | 3.2 | 1 | 6.2 | 0 | 63 |
| HFotsA | 12.0 | 0 | 59.9 | 0 | 611 |
| Hnew | 2.1 | 0 | 4.7 | 0 | 27 |
| totHigh | 1 | 59.6 | 0.0 | 611 | 611 |
| FotsA | 2.2 | 0 | 4.5 | 0 | 38 |
| New | 4.2 | 1 | 9.7 | 0 | 99 |
| totNHigh | 6.5 | 4 | 13.2 | 0 | 137 |

**Table 2. Summary of number of projects and percentages for categorical variables of Tukutuku database**

| Variable | Level | Num. Projects | % Projects |
|---|---|---|---|
| TypeProj | Enhancement | 128 | 65.6 |
| | New | 67 | 34.4 |
| DocProc | No | 104 | 53.3 |
| | Yes | 91 | 46.7 |
| ProImpr | No | 105 | 53.8 |
| | Yes | 90 | 46.2 |
| Metrics | No | 130 | 66.7 |
| | Yes | 65 | 33.3 |

The variable names have the following meaning:

- *nlang*: Number of programming languages adopted in the project.

- *DevTeam*: Number of Developers involved in the project.
- *TeamExp*: Mean number of years of experience for the team members.
- *TotEff*: Effort in person-hours.
- *TotWP*: Total number of Web pages (new and reused).
- *NewWP*: Total number of new Web pages.
- *TotImg*: Total number of images (new and reused).
- *NewImg*: Total number of new images.
- *Fots*: Number of features/functions reused without any adaptation.
- *HFotsA*: Number of reused high-effort features/ functions adapted.
- *Hnew*: Number of new high-effort features/ functions.
- *totHigh*: Total number of high-effort features/ functions.
- *FotsA*: Number of reused low-effort features adapted.
- *New*: Number of new low-effort features/functions.
- *totNHigh*: Total number of low-effort features/ functions.

# 3. The Employed Techniques

In this section, we describe Support Vector Regression (SVR), Tabu Search (TS), and how we have combined them for effort estimation (TS+SVR).

## 3.1 Support Vector Regression

SVR is a regression technique based on Support Vector Machines (SVMs), a very effective machine learning approach [31][33] whose mathematical formulation is inserted in the statistical machine learning theory developed by Vapnik and Chervonenkis [32].

When SVM is used for binary classification, it looks for the hyperplane which separates the elements of the two considered classes with the largest margin, by solving a problem of constrained optimization, where a different constraint is introduced for each of the labeled training set points. Since in some cases it is not possible to find a hyperplane satisfying all constraints, in 1995, Cortes and Vapnik [10] defined a modified version of the approach, by introducing a parameter C to weight the cost of constraint violations: this version is usually referred to as soft margin SVM. As we see in the following, C is one of the parameters which need to be set in the application of SVM based tools.

The solution of this optimization problem can be done in the primary or in the dual space. In the latter case, the introduction of Lagrange multipliers shows how the solution can be expressed as a linear combination of some of the training examples, namely the support vectors. At the same time, the introduction of Lagrange multipliers and dual space solution makes also the adoption of kernel functions [16] particularly smooth, offering a very effective way to face non-linearity. Indeed, a kernel function corresponds to a dot product in the feature space, where the number of features can be much larger than the number of dimensions of the input space (in principle even infinite), and when the number of features increases, the problem is more likely to be linearly separable. Therefore, with the kernel trick, linear classifiers, such as those based on SVMs, can be applied also to non-linear problems.

About the kernel functions, there are two possible solutions: to define an ad-hoc function, suitable for a specific data set/problem, or to use kernels of more general application. Among these, the Radial Basis Function (RBF) is one of the most widely adopted.

When dealing with a regression problem rather than a classification one, we consider the application of the support vector approach to function estimation. In this case, the technique is known as Support Vector Regression (SVR). Indeed, SVR aims at using an approach based on support vectors to find a function which emulates the training set points with an error on each point lower than a constant, usually known as $\varepsilon$. As SVM used for classification, a soft margin version of the algorithm can be used, where the constant $C$ weights the errors larger than $\varepsilon$. A good introduction to SVM/R's mathematical background, some extensions, implementations, and a list of references are given in [28]. To apply support vector techniques in the field of effort estimation, some further preliminary considerations are needed. First of all, in effort estimation the input space consists of the attributes quantifying the cost drivers and the target function is an effort estimate. Thus, it turns out to be a regression problem, and we will deal with SVR and not with SVM.

Moreover, to apply SVR to effort estimation, the two following aspects have to be carefully considered:

1. Large differences in the ranges of the features' values can have the unwanted effect of giving greater importance to some characteristics than to others. To address this issue, a data preprocessing step should be applied.
2. Usually data in the input space are non-linear, and thus kernel functions should be considered to map a problem in a feature space where the target function consists of a line.

Given these two aspects, they can be addressed using many different solutions. An overview is provided in [8][9].

Logarithmic preprocessing is usually adopted since it reduces ranges and at the same time it tackles the linearity issue. This is a typical approach in the field of effort estimation [3][11][13][19].

Summarizing, SVR was applied using two different arrangements of data, namely raw data and logarithmic transformation.

### *3.1.1 Kernel Application and Parameters Setting*

Within the context of this work, we were interested in understanding also the impacts of the adoption of a kernel, which may introduce some other parameters to set. Therefore we chose to stress the Linear and RBF kernels, since they provided the best results in the study presented in [8][9]. The Linear kernel has no specific parameters to set, while the RBF one, implemented in the SVM-light software[1], which was the software used in all the SVR analyses we conducted, is defined as follows:

$$k(u,v) = \exp(-\gamma \, |u - v|^2). \qquad (1)$$

Thus, the actual application of SVR requires the choice of values for the parameters, which are of two different types:

1. parameters of the support vector algorithm, namely C and $\varepsilon$;
2. kernel parameters, which are specific of each kernel, namely $\gamma$ for the RBF.

Since the choice of parameters can have a very strong impact on the application of the SVR technique, they must be set carefully [4][17]. Indeed, the selection of the optimal parameter for a given

1 SVM-light is freely available in http://svmlight.joachims.org/ for scientific use.

data set is an important step to configure SVR since it can cause significant differences in performance. In [8][9] we adopted a semi-automatic approach that allowed us to explore a wide range of variables' values (employing various nested cycles with small incremental steps). For each run, depending on the kernel, the number of executions ranged from some dozens to more than 4000 executions. To evaluate the goodness of the parameter settings, in each of these executions we performed an inner leave-one-out cross validation[2] on the training set (so each cycle of execution required a number of iterations corresponding to the cardinality of the training set) and for each iteration we evaluated the precision on some error summary indicators, which are all detailed in the next section. Thus we chose the setting presenting the lowest error values.

Although such optimization strategy explored a quite large set of parameter combination, it proceeded by brute force, by predefined steps, and did not try to exploit at best the information collected when performing the following steps. Moreover, it is computationally too expensive. Smarter optimization strategies, on the contrary, use all possible clues to focus the search in the most promising areas. Among such strategies, we have investigated a heuristic method to search for the best parameter settings, i.e. TS, which is described in the next section. One of the strength points of TS strategy is that it uses information both in a positive way, to focus the search, and in a negative way, to avoid yet explored area and loops.

## 3.2 Tabu Search

Tabu Search (TS) is an optimization method proposed originally by Glover to overcome some limitations of Local Search [15]. It is a meta-heuristic relying on adaptive memory and responsive exploration of the search space. To apply TS we have to perform the following steps:

- defining a representation of possible solutions and the way to generate the initial one;
- defining local transformations (i.e., moves) to apply to the current solution in order to explore the neighbor solutions;
- choosing a means to evaluate the neighborhood (i.e., an objective function);
- defining the Tabu list size, the aspiration criteria, and the termination criteria.

For searching an optimal SVR parameter setting, a solution $S$ is represented by two parameters for Linear Kernel ($C$ and $\varepsilon$) and by three parameters for the RBF kernel ($C$, $\varepsilon$, and $\gamma$). Since we employed real values to encode such a solution, the search space of TS was represented by all the possible solutions that can be generated assigning the values for $C$, $\varepsilon$, and $\gamma$. An initial solution was generated by randomly choosing the values for each parameter in a defined range. In particular the limits of $C$, $\varepsilon$, and $\gamma$ are (0, 1000], (0, 0.1], and (0, 0.1], respectively. Starting from the current solution, at each iteration the method applied local transformations (moves), defining a set of 25 neighboring solutions in the search space. Each neighbor of a given solution $S$ was defined as a solution obtained by a random variation of it. In particular, a move consisted in changing each parameter of $S$ with probability 0.5; the new parameter was calculated by applying an arithmetic operator, chosen randomly in the range {+, *, - , /}, to $S$ and a number $r$, randomly chosen.

The current solution was compared with the neighboring solutions, to decide whether or not a move to a neighboring solution had to be performed. A number of accuracy measures can be used to compare effort estimation models. All are based on the residual, i.e. the difference between the predicted and actual effort. Among them, we used as objective function a combination of two widely summary measures: the Mean Magnitude of Relative Error (MMRE) and the Mean Magnitude of Relative Error relative to the Estimate (MEMRE) [7], namely the mean of MMRE and MEMRE, whose definitions are reported in the next section. In particular, if the objective function value achieved by a neighboring solution was less than the one achieved by the original solution, the latter was replaced and the neighboring solution was used in the next iteration to explore a new neighborhood. Otherwise the search continues by generating other moves starting from the original solution. To avoid loops and to guide the search far from already visited portions of the search space, the recently visited solutions was marked as "taboo" and stored in a Tabu list. Since only a fixed and fairly limited quantity of information was usually recorded in the Tabu list [13], we prohibited the use of a taboo move for seven iterations. Thus, at each iteration, the Tabu list contained at most seven taboo equations. In order to allow one to revoke taboo, we employed the most commonly used aspiration criterion, namely we permitted a taboo move if it resulted in a solution with an objective function value (i.e. mean of MMRE and MEMRE) better than the one of the current best solution. The search was stopped after a fixed number of iterations (i.e., 100) was reached.

# 4. VALIDATION METHOD AND EVALUATION CRITERIA

To assess the effectiveness of the proposed estimation techniques in predicting development effort, we adopted a hold-out cross-validation method. According to this approach, the original sample is randomly split into two distinct sets, namely training and validation sets. The model learned/constructed using the training set is used to predict the dependent variable of the elements in the validation set. The errors from applying this model are accumulated using summary statistic measures (described in the following). Such hold out methodology assumes the experimental conditions to remain constant, that is, that there is not a temporal evolution which causes the experimental conditions to change. In the final application, data collected in the past will be used to build an estimator able to solve problems in the future, and such procedure is effective only as long as the overall contextual conditions remain sufficiently constant. In such constant conditions, a hold-out validation is definitely fair, since the observations included in the validation set are completely different from those used to obtain the estimates. It is our view that considering only one split into training and validation sets, even if randomly selected, can be misleading, as it can correspond to either an extremely advantageous or, on the contrary, disadvantageous, split. Therefore, in our case study, we employed two random splits, so that the probability of having such extreme

[2]In a leave-one-out cross-validation, a single observation from the original sample is used to evaluate the model that is trained using the remaining observations. This is repeated until each observation in the sample is used once as validation data. The application of a leave-one-out cross validation on the training set has allowed us to prevent problems of model overfitting that could hinder the model having good prediction accuracy on an out data set.

cases in both experiments is highly reduced. Furthermore, to compare the results obtained by combining TS and SVR with those obtained in our previous studies [8][9][20], these two splits were the same employed by Mendes in [20], where the two training sets were obtained by randomly selecting two times 130 observations from the original 195 projects contained in the Tukutuku database, while the remaining 65 observations were included in the validation set.

To assess the accuracy of the obtained estimations, we employed some commonly used measures for effort estimation, such as the Mean of Magnitude of Relative Error (MMRE), Median of MRE (MdMRE), and Prediction at level 25% (Pred(25)) [7]. Let us recall that Pred(*l*) measures the percentage of estimates that are within *l*% of the actual values, and *l* is usually set at 25. MRE is the basis for calculating MMRE and MdMRE, and is defined as:

$$MRE = |e - \hat{e}| / e \quad (2)$$

where $e$ represents actual effort and $\hat{e}$ estimated effort. The difference between MMRE and MdMRE is that the former is more sensitive to predictions containing extreme MRE values. Note that Kitchenham *et al.* [17] showed that MMRE and Pred(*l*) are respectively measures of the spread and kurtosis of *z*, where ($z = |\hat{e}|/e$). They suggested the use of boxplots of *z* and boxplots of the residuals ($e-\hat{e}$) as useful complements to simple summary measures, since they can give a good indication of the distribution of residuals and *z*, and can help gathering insight on summary statistics such as MMRE and Pred(*l*). Indeed, boxplots are widely employed in exploratory data analysis since they provide a quick visual representation to summarize the data, using five values: median, upper and lower quartiles, minimum and maximum values, and outliers [17]. The same authors suggest also the use of the Magnitude of Relative Error relative to the Estimate (EMRE) as a comparative measure as well [18]. The EMRE has the same form of MRE, but the denominator is the estimate, giving thus a stronger penalty to under-estimates.

$$EMRE = |e - \hat{e}| / \hat{e} \quad (3)$$

As with the MRE, we can also calculate the mean EMRE (MEMRE) and Median EMRE (MdEMRE).

Finally, to verify if the differences observed using the above measures were legitimate or due to chance, we checked if the absolute residuals obtained with the application of the various estimation techniques come from the same population. If they do, it means that there are no significant differences between the data values being compared. We accomplished the statistical significance test using the nonparametric Wilcoxon Signed Rank test, with $\alpha = 0.05$ [18][26].

## 5. RESULTS AND DISCUSSION

In this section we present the results obtained when applying SVR in combination with TS (TS+SVR) with the different settings detailed in Table 3. We used this naming convention to facilitate the reader compare current results with the ones presented in [9].

As described in Section 3, we used TS to identify the best parameter settings for each SVR configuration. Such parameters are related to the SVR technique (namely $C$ and $\varepsilon$), and to the specific kernel (namely $\gamma$ for the RBF kernel). TS was applied on each of the two training sets, leading to the identification of the values of the parameters reported in Table 6. In addition, Table 7 shows the values of the parameters obtained in [8] where we did not consider TS. Please note that these tables, together with 8 and 9, are at the end of the paper.

**Table 3. The considered data configurations and kernels**

| Data Configuration | Variable transformation | Kernel |
|---|---|---|
| C1 | No transformation of the features | Linear / RBF |
| C3 | Log transformation of the features | Linear / RBF |

Table 8 shows the results in terms of MMRE, MdMRE, Pred(25), MEMRE, and MdEMRE for each configuration adopted and each split obtained by applying TS+SVR.

When variables are not transformed (i.e., configuration C1), the best results in terms of summary measures are obtained by the RBF kernel for the first validation set and by the Linear kernel for the second validation set. In order to verify if the differences observed using these measures were legitimate or due to chance, we checked the statistical significance of the results by comparing the absolute residuals obtained with the different settings using the nonparametric test Wilcoxon Signed Ranks test ($\alpha = 0.05$). Results showed that Linear and RBF kernels were comparable for both validation sets. With regard to the Log transformation-based configurations (i.e., C3), the best results in terms of summary measures were obtained using the RBF kernel. These results were supported by the Wilcoxon test (i.e., the absolute residuals obtained with RBF kernel are significantly smaller than those achieved with Linear kernel).

In order to verify the effectiveness of the combined use of TS and SVR, in Table 9 we showed the results obtained in [9] where SVR was used without TS. For both validation sets, SVR and TS+SVR presented comparable results when Linear kernel was used (i.e., C1(Lin) and C3(Lin)), while the introduction of TS provided better results when RBF kernel was employed (i.e., C1(RBF) and C3(RBF)). Moreover, the Wilcoxon Signed Ranks test revealed that, for both validation sets, in the case of C1(RBF) the absolute residuals obtained with the use of TS were significant lower than those obtained without using it. Furthermore, for the second validation set the use of TS also provided significant better results in the cases of C3(RBF) and C1(Lin). Thus, the use of TS to configure SVR has allowed us to significantly improve the accuracy of the obtained estimates. This observation, together with the fact that TS allows us to automate the configuration parameter settings, encourages us to further investigate TS to improve the effectiveness of estimation techniques.

Once we have investigated how the various configurations (obtained by applying TS or not) performed, we were interested in understanding which configuration provided the best results overall. Again we took into account the same summary measures and the boxplots of absolute residuals. The analysis of summary statistics suggested that the best results were achieved with C3 employing the RBF kernel for both validation sets. All these results were confirmed by the boxplots in Figure 1 and Figure 2. In particular, the boxplot of C3(RBF) has the box length and the tails less skewed than those of the other boxplots. Furthermore, the median of the boxplot of C3(RBF) is closer to zero than all the others.

In order to verify if the differences observed using summary measures and boxplots of absolute residuals were legitimate or due to chance, we also checked the statistical significance of the results by applying the Wilcoxon test on the absolute residuals. The results for the first and the second validation sets are reported in Table 4 and Table 5, respectively, where "Yes" in a cell means that the technique indicated on the row is significantly superior to the one indicated on the column. Regarding the first validation set, the results show that the absolute residuals based on C3 using the RBF kernel are significantly smaller than those obtained by using the other settings (p-values < 0.014). The second validation set is characterized by similar results (p-values < 0.07). Thus, we can conclude that the Log transformation-based configurations (i.e., C3) provide the best results when RBF kernel is employed.

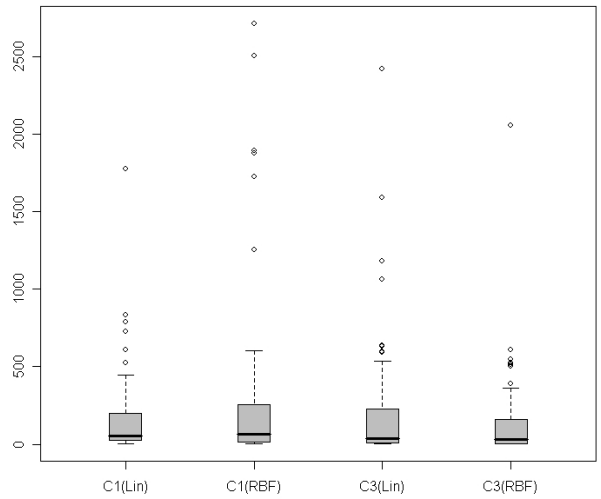


**Figure 1. Boxplots of abs residuals for the 1st validation set**

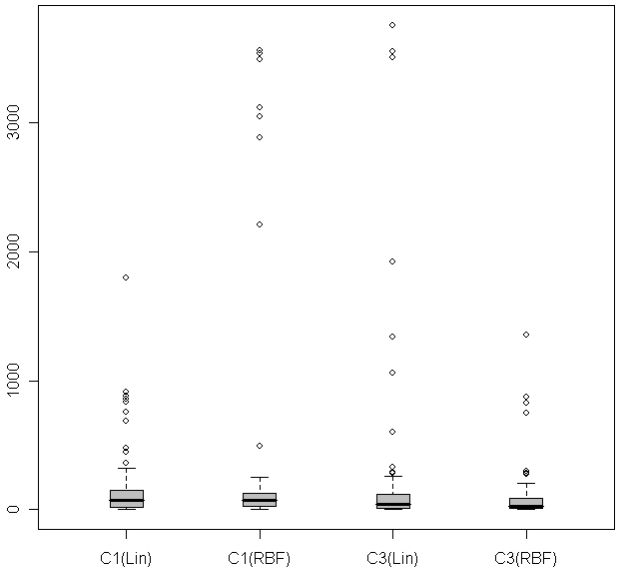


**Figure 2. Boxplots of abs residuals for the 2nd validation set**

From all these considerations, we can for sure positively answer to our first question "Is Tabu Search able to effectively set Support Vector Regression configuration parameters?"

## 5.1 Comparison with other techniques

It is worth noting that the estimates obtained with SVR in [8][9] were significantly superior to those obtained in [20] with widely used estimation techniques (namely, with MSWR and CBR) and BNs, as used in the previous case study, employing the same validation sets. Furthermore, the estimates obtained with SVR were also significant better than those obtained with Mean and the Median value of the effort in the data set.

We have performed the same comparisons, taking into account configuration C3 with RBF kernel, since the analysis presented in the previous section has revealed that this configuration provided better results with respect to the others. Again, we used the results obtained in [20]. The summary measures for the two splits are reported in Table 9 and Table 10, where we have adopted the following acronyms:

- MSWR: Manual Stepwise Regression
- CBR1: Case-based reasoning using one analogy
- CBR2: Case-based reasoning using two analogies;
- CBR3: Case-based reasoning using three analogies;
- MeanEffort: Mean value of the effort in the data set;
- MedianEffort: Median value of the effort in the data set.

The summary measures MMRE, MdMRE, Pred(25), MEMRE, and MdEMRE suggest that, for both validation sets, TS+SVR provided better results than any other technique, even though they do not fit the thresholds suggested by Conte *et al.* [7]. Moreover, even if all the techniques outperform MeanEffort, only TS+SVR and MSWR are characterized by better predictions than MedianEffort.

**Table 4. Comparison of the absolute residuals obtained with the different SVR settings using Wilcoxon test (p-value between brackets) for the first validation set**

| | C1(Lin) | C1(RBF) | C3(Lin) | C3(RBF) |
|---|---|---|---|---|
| C1(Lin) | - | No (0.055) | No (0.836) | - |
| C1(RBF) | - | - | - | - |
| C3(Lin) | - | **Yes (0.031)** | - | - |
| C3(RBF) | **Yes (0.014)** | **Yes (0.000)** | **Yes (0.001)** | - |

**Table 5. Comparison of the absolute residuals obtained with the different SVR settings using Wilcoxon test (p-value between brackets) for the second validation set**

| | C1(Lin) | C1(RBF) | C3(Lin) | C3(RBF) |
|---|---|---|---|---|
| C1(Lin) | - | No (0.253) | No (0.925) | - |
| C1(RBF) | - | - | No (0.918) | - |
| C3(Lin) | - | - | - | - |
| C3(RBF) | **Yes (0.000)** | **Yes (0.001)** | **Yes (0.007)** | - |

Figure 3 and Figure 4 show respectively the boxplots of absolute residuals and of z, obtained for the first test set. The analysis of the boxplots of absolute residuals confirmed the patterns abovementioned. Indeed, the TS+SVR's distribution is less skewed than the other distributions. Furthermore, the outliers of SVR's boxplot are closer to the tails than the outliers for the other techniques, and the median of TS+SVR boxplot is closer to zero. Similar considerations can be done for the boxplots of z, even if

MSWR, BNHyHu, and MedianEffort have median very close to TS+SVR, and box length and tails less skewed. However, the outliers of TS+SVR are less far from zero then those of the others.

As for the second validation set, the boxplots of absolute residuals and z are shown in Figure 5 and Figure 6, supporting the results abovementioned. Indeed, the median of the boxplots of absolute residuals for TS+SVR was closer to zero than the other medians. The boxplot of MedianEffort was very similar to the one of TS+SVR; however, the outliers of the boxplot of TS+SVR are closer to the tails with respect to the one of MedianEffort. On the other hand, the boxplots of z suggested that TS+SVR provided better results than MedianEffort, and the boxplots of TS+SVR and MSWR were very similar. Indeed, the median of the TS+SVR and MSWR boxplots are closer to zero and these boxplots are less skewed than the others. Furthermore, the outliers of the TS+SVR and MSWR are closest to the tails.

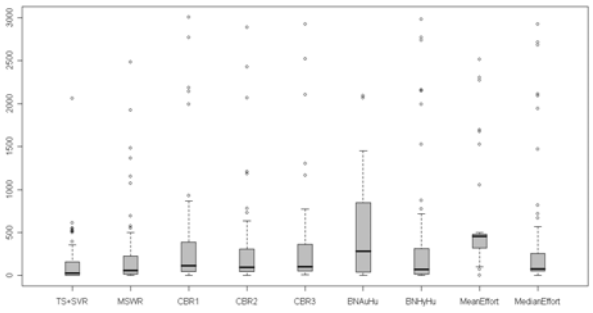


**Figure 3. Boxplots of abs residuals for the 1st validation set**

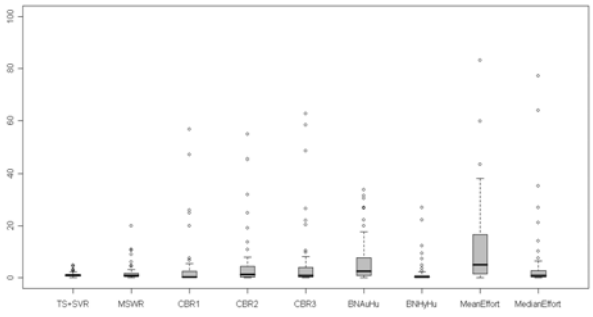


**Figure 4. Boxplots of z obtained for the 1st validation set**

In order to verify if there are statistically significant differences among the techniques, again, we applied the Wilcoxon test ($\alpha$ = 0.05). The results, reported in Table 11 and Table 12, reveal that the predictions obtained with TS+SVR are significantly superior to those obtained by all the other techniques, on both the validation sets.

From all these considerations, we can for sure positively answer also to our second research question: "Are the effort predictions obtained by the combination of Tabu Search and Support Vector Regression significantly superior to the ones obtained by other techniques?"

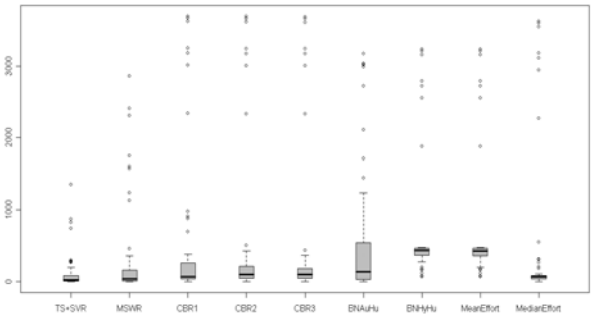


**Figure 5. Boxplots of abs residuals for the 2nd validation set**

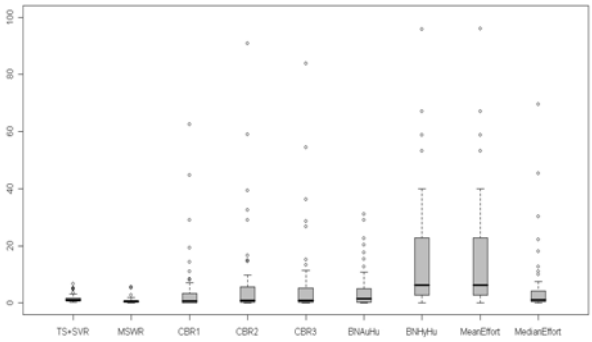


**Figure 6. Boxplots of z obtained for the 2nd validation set**

## 6. RELATED WORK

Several studies have investigated and compared the effectiveness techniques such as Linear Regression (LR), Stepwise Regression (SWR), Case-Based Reasoning (CBR), and Regression Trees (RT), in estimating Web application development effort [11][13][25]. For instance, in [11] all these techniques where applied in combination with two sets of size measures. The first one included some length measures (e.g., number of Web pages, number of server-side script and applications, etc.) while the second contained the components used to evaluate the Web Objects measure. The results revealed that CBR provided better estimates with the first set, and LR for the second one, but no statistically significant differences were found among the residuals of the two sets of measures.

About the use of SVR, to the best of our knowledge, two previous studies investigated the application of this technique for effort estimation. Oliveira [27], who was the first to apply SVR to this field, used data from 18 applications from the well-known NASA software project data set [1]. By using a leave-one-out cross-validation, the author reported that SVR significantly outperformed both LR and Radial Basis Function Networks (RBFNs), in terms of the indicators MMRE and Pred(25). In the second study, the authors proposed a machine learning-based method that provided an effort estimate and corresponding confidence interval [2]. To assess the defined method, they performed a case study using the Desharnais [12] and NASA [1] data sets. The results of this empirical analysis showed that the proposed method was characterized by better performance with

respect to the previous study. It is worth noting that none of these two studies used data from cross-company data sets, thus setting the motivation for and contribution of our work.

About the use of meta-heuristics for parameter setting, some efforts have been made to employ Genetic Algorithms (GA) to improve the estimation performance of existing estimation techniques. The first attempt to combine evolutionary approaches with an existing effort estimation technique was made by Shukla [29] applying genetic algorithms to Neural Networks (NN) predictor (namely, neuro-genetic approach, GANN) in order to improve its estimation capability. Results were significantly better than other techniques, such as a modified version of the RTs.

More recently, Chiu and Huang applied GA to CBR [5] on two datasets, but even if the results were positive, they were comparable with LR and RT.

About Tabu Search, to the best of our knowledge, only one case study was performed to assess its use for estimating software development effort. In particular, Ferrucci *et al.* carried out a preliminary case study by applying TS on Desharnais data set, obtaining interesting results, motivating further investigation on search-based methods [14].

Finally, about other studies on the Tukutuku dataset, in [13] they were compared the two sets of measures used in [11], and the Tukutuku one [22]. The empirical results showed that all the measures provided good estimations in terms of MMRE, MdMRE, and Pred(25) and the study largely confirmed the results of previous works. It is worth noting that the study presented in [25] employed 37 Web applications developed by postgraduate students while the studies reported in [11][13] were based on 15 industrial Web applications. Recently, Mendes and Mosley investigated the use of Bayesian Networks (BN) for Web effort estimation using the Tukutuku database [20][23]. In particular, they built eight BNs by exploiting both automatic tools, such as Hugin and PowerSoft, and a causal graph elicited by a domain expert with parameters automatically fit using the same training sets used in the automated elicitation (thus working as composite models) [6]. As in our study, they employed the Tukutuku database containing data on 195 Web projects and compared the accuracy of the obtained estimates with those obtained using Manual SWR (MSWR), CBR, median and mean effort. That analysis revealed that MSWR provided significantly better estimations than any of the models obtained using BNs and was the unique approach that provided significantly better results than the median effort model.

# 7. CONCLUSIONS

In the paper, we have assessed whether the use of Tabu Search can be effective for configuring Support Vector Regression parameters in estimating Web application development effort. To this end, we have applied TS+SVR to a cross-company data set of Web applications, the Tukutuku database. In particular, we explored and compared the SVR configurations that provided the best results in two previous case studies on the same data set [8][9].

The results of the empirical analysis have highlighted that applying the meta-heuristic Tabu Search, we obtained a significant improvement on performances. While we are not guaranteed that this is the global optimum, the results we are presenting are even better than the ones discussed in [8][9], and represent the best performance ever obtained on the problem at the hand, up to now. Of course, the experimental results here presented hold only with respect to the Tukutuku database and they should be assessed on further data as soon as they are available. However, these results, together with those of previous case studies [8][9], are surely interesting enough to suggest the use of SVR among the techniques that are suitable for Web development effort estimation using a cross-company database. Moreover, they suggest that Tabu Search can be effectively used to configure SVR parameters.

# 8. ACKNOWLEDGMENTS

Authors wish to thank all companies that volunteered data to the Tukutuku database. The research has also been carried out exploiting the computer systems funded by University of Salerno's Finanziamento Medie e Grandi Attrezzature (2005) for the Web Technologies Research Laboratory.

# 9. REFERENCES

[1] J.W. Bailey, V.R. Basili "A meta model for software development resource expenditure", Procs. International Conference on Software Engineering, San Diego, California, USA, 1981, pp. 107–116.

[2] P. L. Braga, A. L. I. Oliveira, S. R. L. Meira "Software Effort Estimation using Machine Learning Techniques with Robust Confidence Intervals", HIS 2007: 352-357.

[3] L. Briand, T. Langley, I. Wiekzorek, "A Replicated Assessment and Comparison of Common Software Cost Modeling Techniques", Procs. International Conference on Software Engineering, IEEE press, 2000, pp. 377–386.

[4] K. Chen, C. Wang, "Support vector regression with genetic algorithms in forecasting tourism demand", Tourism Management 28 (2007), 215–226

[5] N.-H. Chiu, S-. Huang, "The adjusted analogy-based software effort estimation based on similarity distances", *Journal of Systems and Software* 80(4) (2007), pp. 628–640.

[6] S. Chulani, B. Boehm, B. Steece "Bayesian Analysis of Empirical Software Engineering Cost Models", IEEE TSE 25 (1999) 573–583.

[7] S.D. Conte, H.E. Dunsmore, V.Y. Shen, "Software Engineering Metrics and Models", Benjamin-Cummins, 1986.

[8] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, E. Mendes, "Applying support vector regression for web effort estimation using a cross-company data set", Procs. Empirical Software Engineering and Measurement, IEEE press, 2009, pp: 191-202

[9] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, E. Mendes, "Investigating the use of Support Vector Regression for Web Effort Estimation", accepted for publication in Empirical Software Engineering Journal.

[10] C. Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, 20, 1995

[11] G. Costagliola, S. Di Martino, F. Ferrucci, C. Gravino, G. Tortora, G. Vitiello, "Effort estimation modeling techniques: a case study for web applications", Procs. Intl. Conference on Web Engineering (ICWE'06), 2006, 9-16.

[12] J. M. Desharnais, Analyse statistique de la productivitie des projets in 834 formatique a partie de la technique des point des fonction, Ph.D. thesis, 835 Unpublished Masters Thesis, University of Montreal (1989).

[13] S. Di Martino, F. Ferrucci, C. Gravino, E. Mendes "Comparing Size Measures for Predicting Web Application Development Effort: A Case Study", Procs. Empirical Software Engineering and Measurement, IEEE press, 2007, pp. 324–333.

[14] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, "Using Tabu Search to Estimate Software Development Effort". Procs. International Conferences on Software Process and Product Measurement. LNCS 5891. Springer-Verlag, 2009, pp. 307-320.

[15] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, 1997.

[16] T. Hofmann, B. Scholkopf, A. Smola, "Kernel methods in machine learning" Annals of Statistics, 36(3), 2008, 1171–1220.

[17] S. Keerthi, "Efficient tuning of SVM hyper-parameters using radius/margin bound and iterative algorithms". IEEE Transaction on Neural Networks, 13(5), (2002), 1225–1229.

[18] B. Kitchenham, L. M. Pickard, S. G. MacDonell, M. J. Shepperd "What accuracy statistics really measure", IEE Proceedings Software 148 (3) (2001) 81–85.

[19] B. A. Kitchenham, E. Mendes, "A Comparison of Cross-company and Single-company Effort Estimation Models for Web Applications", Procs. EASE 2004, 2004, pp. 47-55.

[20] E. Mendes "The Use of Bayesian Networks for Web Effort Estimation: Further Investigation", Procs. International Conference on Web Engineering (2008).

[21] E. Mendes, S. Counsell, "Web Development Effort Estimation using Analogy", Procs. Australian Software Engineering Conference, pp. 203-212, 2000.

[22] E. Mendes, N. Mosley, S. Counsell, "Investigating Web Size Metrics for Early Web Cost Estimation", Journal of Systems and Software, 77 (2), 157-172, 2005.

[23] E. Mendes, N. Mosley "Bayesian Network Models for Web Effort Prediction: A Comparative Study", IEEE TSE 34 (6) (2008) 723–737.

[24] E. Mendes, N. Mosley, S. Counsell, "Early Web Size Measures and Effort Prediction for Web Costimation", Procs. IEEE Metrics Symposium, pp. 18-29, 2003.

[25] E. Mendes, N. Mosley, S. Counsell, "Comparison of Length, complexity and functionality as size measures for predicting Web design and authoring effort", IEE Procs. Software 149 (3) 86-92, 2002.

[26] E. Mendes, S. D. Martino, F. Ferrucci, C. Gravino "Cross-company vs. single-company web effort models using the Tukutuku database: An extended study", Journal of System & Software 81 (5) (2008) 673–690.

[27] A. L. I. Oliveira, "Estimation of software project effort with support vector regression", Neurocomputing, 69(13-15):1749–1753, 2006.

[28] B. Scholkopf, A. Smola, "Learning with Kernels". 2002, MIT Press

[29] K. K. Shukla, "Neuro-genetic prediction of software development effort", *Information and Software Technology* 42 (10) (2000), pp. 701–713.

[30] A. J. Smola , B. Schölkopf, "A tutorial on support vector regression", Statistics and Computing, 14 (3) 2004, 199-222.

[31] V. Vapnik, A. Lerner, "Pattern recognition using generalized portrait method", Automation and Remote Control 24, 1963, 774-780.

[32] V. Vapnik, A. Chervonenkis, "A note on one class of perceptrons", Automatics and Remote Control 1964, 25.

[33] V. Vapnik," The nature of Statistical Learning Theory", Springer-Verlag, 1995

**Table 6. SVR parameter settings obtained on the two training sets using TS**

| | **First training set** | | | | **Second training set** | | | |
|---|---|---|---|---|---|---|---|---|
| Kernel | Linear | | RBF | | Linear | | RBF | |
| Conf. | C1 | C3 | C1 | C3 | C1 | C3 | C1 | C3 |
| C | 45.345 | 39.255 | 356.963 | 9.139 | 7.606 | 1.454 | 84.95 | 2.165 |
| EPS | 0.01 | 0.013 | 0.037 | 0.01 | 0.087 | 0.003 | 0.026 | 0.096 |
| Gamma | - | - | 0.003 | 0.04 | - | - | 0.016 | 0.024 |

**Table 7. SVR parameter settings obtained on the two training sets without using TS**

| | **First training set** | | | | **Second training set** | | | |
|---|---|---|---|---|---|---|---|---|
| Kernel | Linear | | RBF | | Linear | | RBF | |
| Conf. | C1 | C3 | C1 | C3 | C1 | C3 | C1 | C3 |
| C | 51 | 63 | 50 | 2 | 151 | 0.02 | 63 | 2 |
| EPS | 0.01 | 0.0001 | 0.01 | 0.001 | 0.01 | 0.0001 | 0.01 | 0.001 |
| Gamma | - | - | 0.01 | 0.01 | - | - | 0.01 | 0.2 |

**Table 8. Accuracy measures obtained by using the parameters settings given by Tabu Search**

| | First validation set | | | | | Second validation set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **MMRE** | **MdMRE** | **Pred(25)** | **MEMRE** | **MdEMRE** | **MMRE** | **MdMRE** | **Pred(25)** | **MEMRE** | **MdEMRE** |
| C1(Lin) | 1.989 | 0.582 | 0.234 | 1.357 | 0.732 | 2.443 | 0.700 | 0.231 | 2.209 | 0.716 |
| C1(RBF) | 1.712 | 0.663 | 0.297 | 1.397 | 0.735 | 2.733 | 0.954 | 0.200 | 2.855 | 0.692 |
| C3(Lin) | 1.151 | 0.552 | 0.281 | 1.122 | 0.530 | 1.007 | 0.539 | 0.231 | 0.764 | 0.491 |
| C3(RBF) | 0.590 | 0.339 | 0.391 | 0.689 | 0.365 | 0.856 | 0.455 | 0.400 | 0.498 | 0.410 |

**Table 9. Accuracy measures obtained by using the parameters settings given by the semi-automatic approach in [9]**

| | First validation set | | | | | Second validation set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **MMRE** | **MdMRE** | **Pred(25)** | **MEMRE** | **MdEMRE** | **MMRE** | **MdMRE** | **Pred(25)** | **MEMRE** | **MdEMRE** |
| C1(Lin) | 1.984 | 0.583 | 0.234 | 1.355 | 0.735 | 2.298 | 0.826 | 0.077 | 2.137 | 0.754 |
| C1(RBF) | 1.422 | 0.777 | 0.203 | 2.979 | 0.733 | 2.723 | 0.751 | 0.108 | 7.804 | 0.886 |
| C3(Lin) | 1.151 | 0.555 | 0.281 | 1.118 | 0.555 | 1.215 | 0.45 | 0.338 | 0.68 | 0.542 |
| C3(RBF) | 0.591 | 0.411 | 0.344 | 0.603 | 0.467 | 0.91 | 0.36 | 0.415 | 0.506 | 0.413 |

**Table 10. Accuracy measures obtained by using the considered estimation techniques**

| | First validation set | | | | | Second validation set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **MMRE** | **MdMRE** | **Pred(25)** | **MEMRE** | **MdEMRE** | **MMRE** | **MdMRE** | **Pred(25)** | **MEMRE** | **MdEMRE** |
| TS+SVR | 0.590 | 0.339 | 0.391 | 0.689 | 0.365 | 0.856 | 0.455 | 0.400 | 0.498 | 0.410 |
| MSWR | 1.50 | 0.64 | 0.23 | 1.36 | 0.64 | 0.73 | 0.66 | 0.11 | 2.86 | 1.21 |
| CBR1 | 5.27 | 0.97 | 0.08 | 31.70 | 3.43 | 4.46 | 0.92 | 0.08 | 21.81 | 0.95 |
| CBR2 | 5.06 | 0.87 | 0.11 | 3.59 | 0.81 | 6.73 | 0.89 | 0.15 | 15.65 | 0.90 |
| CBR3 | 5.63 | 0.97 | 0.09 | 4.17 | 0.88 | 6.09 | 0.84 | 0.09 | 13.26 | 0.89 |
| BNAuHu | 7.65 | 1.67 | 7.69 | 1.07 | 0.76 | 4.09 | 0.96 | 0.02 | 7.90 | 0.93 |
| BNHyHu | 1.90 | 0.86 | 0.15 | 13.06 | 2.38 | 27.95 | 5.31 | 0.09 | 1.34 | 0.90 |
| Mean Effort | 30.35 | 3.99 | 15.38 | 1.07 | 0.91 | 27.94 | 5.31 | 0.03 | 1.34 | 0.90 |
| Median Effort | 5.02 | 0.93 | 0.09 | 4.43 | 0.94 | 4.95 | 0.89 | 0.15 | 4.62 | 0.78 |

**Table 11. Comparison of the absolute residuals obtained for the first validation set, using the Wilcoxon test (p-values are reported between brackets)**

| | MSWR | CBR1 | CBR2 | CBR3 | BNAuHu | BNHyHu | Mean Effort | Median Effort |
|---|---|---|---|---|---|---|---|---|
| TS+SVR | **Yes (0.044)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.025)** | **Yes (0.000)** | **Yes (0.000)** |
| MSWR | - | **Yes (0.000)** | **Yes (0.002)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** |
| CBR1 | | - | Yes (0.052) | No (0.398) | No (0.288) | | **Yes (0.022)** | |
| CBR2 | | | - | No (0.227) | **Yes (0.022)** | No (0.229) | **Yes (0.000)** | No (0.335) |
| CBR3 | | | | - | **Yes (0.023)** | No (0.288) | **Yes (0.000)** | No (0.422) |
| BNAuHu | | | | | - | | No (0.223) | No (0.822) |
| BNHyHu | | **Yes (0.000)** | | | **Yes (0.038)** | - | **Yes (0.022)** | |
| Mean Effort | | | | | | | - | |
| Median Effort | | **Yes (0.003)** | | | | **Yes (0.042)** | **Yes (0.002)** | - |

**Table 12. Comparison of the absolute residuals obtained for the second validation set, using the Wilcoxon test (p-values are reported between brackets)**

| | MSWR | CBR1 | CBR2 | CBR3 | BNAuHu | BNHyHu | Mean Effort | Median Effort |
|---|---|---|---|---|---|---|---|---|
| TS+SVR | **Yes (0.046)** | **Yes (0.000)** | **Yes (0.001)** | **Yes (0.001)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.010)** |
| MSWR | - | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.001)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.009)** |
| CBR1 | | - | No (0.909) | No (0.893) | No (0.241) | **Yes (0.009)** | **Yes (0.009)** | |
| CBR2 | | | - | | No (0.380) | **Yes (0.000)** | **Yes (0.000)** | |
| CBR3 | | | **Yes (0.000)** | - | No (0.264) | **Yes (0.000)** | **Yes (0.000)** | |
| BNAuHu | | | | | - | **Yes (0.000)** | **Yes (0.000)** | |
| BNHyHu | | | | | | - | **Yes (0.000)** | |
| Mean Effort | | | | | | | - | |
| Median Effort | | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | **Yes (0.000)** | - |