INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND COMPUTING
© International Science Press, ISSN: 2229-7413

ON PREDICTING SOFTWARE DEVELOPMENT EFFORT USING MACHINE LEARNING TECHNIQUES AND LOCAL DATA

Lukasz Radlinski and Wladyslaw Hoffmann

This paper analyses the accuracy of predictions for software development effort using various machine learning techniques. The main aim is to investigate the stability of these predictions by analyzing if particular techniques achieve a similar level of accuracy for different datasets. Two key assumptions are that (1) predictions are performed using local empirical data and (2) very little expert input is required. The study involves using 23 machine learning techniques with four publicly available datasets: COCOMO, Desharnais, Maxwell and QQDefects. The results show that the accuracy of predictions for each technique varies depending on the dataset used. With feature selection most techniques provide higher predictive accuracy and this accuracy is more stable across different datasets. The highest positive impact of feature selection on the accuracy has been observed for the K technique, which has generated the most accurate predictions across all datasets.*

Keywords: *effort prediction, local data, process factors, machine learning, prediction accuracy*

1. INTRODUCTION

Software development effort can be predicted using various approaches. Some of them require large dataset of past projects while others require strong input from domain expert. Some approaches are easy to use while others are difficult to follow and time consuming. Jørgensen and Shepperd [10] performed a systematic review of software development effort estimation studies. They have provided and analyzed an extensive list of relevant publications in their paper. They have also performed various classifications for this subject.

In this study, we focus on the subfield of this popular research field by attempting to answer the following main question: **Is it possible to easily predict software development effort from local data?**

There are two key constraints limiting the scope of this study: easy prediction and using local data. By easy prediction we mean using such procedure, which can be followed even without having background in quantitative statistical techniques. Some of such techniques can often be properly applied only after performing various checks related to input data, for example about the normality of distributions or linear relationships between variables to name just a couple of the simplest. The procedures of using such techniques may be complex and time-consuming. Thus, in this study we examine a performance of a set of machine learning techniques that can be applied easily to the empirical input data. Naturally, they also require some basic data preparation but this step is very simple.

The second constraint is to use local data. By this, we mean data from a single software company or a software development division of a larger organization. Various researchers have analyzed effort predictions from local and cross-company data. In [12, 16] the authors compare the accuracy of predictions from local and cross-company data, including results achieved in earlier work by other authors. This analysis revealed that in some experiments similar accuracy have been achieved, while in other experiments predictions based on local data have been more accurate than based on cross-company data. When gathering cross-company data there are typically more people involved than when gathering local data. Furthermore, for cross-company data there is a larger risk in different understanding the data by

* Institute of Information Technology in Management, Faculty of Economics and Management, University of Szczecin, Poland, *E-mail: lukrad@uoo.univ.szczecin.pl, woodieh@gmail.com*

people who gather them. For example, the assessment "very high" for team experience is highly subjective not only for different individuals but also from a perspective of specific company, its environment and culture. In addition, there are various ways of calculating project effort depending on the types of activities included, people involved in development, and many others. For these reasons, to reduce the difficulties associated with using cross-company data, we have decided to use only local data of software projects.

To support answering the main research question of this paper we analyze the following set of more detailed questions:

- What is the accuracy of predictions obtained using various machine learning techniques?
- What are the differences between predictions using these techniques?
- What is the impact of feature selection on achieved accuracy of predictions?
- How stable are predictions from particular technique across various datasets?

This paper is organized as follows: Section 2 provides information on the procedure, techniques and datasets used in this experiment. Main points of the data preparation stage have been summarized in Section 3. Section 4 discusses achieved results. An overview of the threats to validity has been included in Section 5. We sum up the whole study in Section 6.

2. RESEARCH APPROACH

2.1. Dataset Used

In this study we have used four publicly available datasets: COCOMO [2, 3], Desharnais [5], Maxwell [15], and QQDefects [6, 7]. They have been summarized in Table 1. All datasets are available in the PROMISE repository [4]. Most of these datasets, i.e. COCOMO, Desharnais, and Maxwell, have been widely used in earlier experiments. Selected results from these studies have been discussed in Section 4.3.

QQDefects has been originally used in [6, 7] to predict the number of defects. In this study we have not used a variable "number of defects" and attempt to use the dataset for effort prediction. However, a slightly modified version of this dataset has been used. This includes an additional predictor "project type" and additional cases for which the data has not been previously published. It should be noted that QQDefects is the only dataset where the number of variables (31) exceeds the number of cases (29).

Tables 2–5 list the variables in particular dataset used in this study. Most original datasets contain other variables. However, since they are not relevant to this study, we have removed them and not listed in these tables. Detailed descriptions of datasets and variables are available in relevant original studies.

We can observe that there are high differences between these datasets, related to the number of cases and number of predictors of specific types. Additionally, projects in different datasets are typically described from different point of view. For example, Maxwell dataset contains data on the development process but is more focused on the nature of projects. On the other hand, in QQDefects dataset almost all variables describe the development process. Furthermore, very few variables have been included in all datasets, or

Table 1
Summary of Datasets Used

Datasets	*Number of cases*	*Project size*	*Effort*	*Number of numeric predictors*	*Number of ordinal predictors*	*Number of nominal predictors*
COCOMO [2, 3]	93	KLOC	person-months	2	15	5
Desharnais [5]	81	function points	person-hours	7	0	1
Maxwell [15]	62	function points	person-hours	3	15	6
QQDefects [6, 7]	29	KLOC	person-hours	1	27	2

Table 2
List of Variables in COCOMO Dataset

Symbol	*Name*	*Type*	*Symbol*	*Name*	*Type*
projectname	name of project	nominal	acap	analysts capability	ordinal
cat	category of application	nominal	aexp	application experience	ordinal
forg	flight or ground system	nominal	pcap	programmers capability	ordinal
center	which NASA center	nominal	vexp	virtual machine experience	ordinal
mode	semidetached, embedded, organic	nominal	lexp	language experience	ordinal
rely	required software reliability	ordinal	modp	modern programing practices	ordinal
data	data base size	ordinal	tool	use of software tools	ordinal
cplx	process complexity	ordinal	sced	schedule constraint	ordinal
time	time constraint for cpu	ordinal	year	year of development	numeric
stor	main memory constraint	ordinal	kloc	number of KLOC	numeric
virt	machine volatility	ordinal	effort	development effort	numeric
turn	turnaround time	ordinal			

Table 3
List of Variables in Desharnais Dataset

Symbol	*Name*	*Type*	*Symbol*	*Name*	*Type*
TeamExp	team experience	numeric	Entities	number of entities	numeric
Language	programming language	nominal	FPNon Adjust	function points non-adjusted	numeric
ManagerExp	manager experience	numeric	Adjustment	function point complexity adjustment factor	numeric
YearEnd	year project ended	numeric	Effort	actual effort	numeric
Transactions	number of transactions	numeric			

Table 4
List of Variables in Maxwell Dataset

Symbol	*Name*	*Type*	*Symbol*	*Name*	*Type*
App	application type	nominal	Complex	software logical complexity	ordinal
Har	hardware platform	nominal	Reqvol	requirements volatility	ordinal
Dba	type of database	nominal	Qualreq	quality requirements	ordinal
Ifc	type of user interface	nominal	Effreq	efficiency requirements	ordinal
Source	where developed	nominal	Instreq	installation requirements	ordinal
Telonuse	Telon use	nominal	Stanskil	staff analysis skills	ordinal
Nlan	number of languages	numeric	Stappknow	staff application knowledge	ordinal
Custpart	customer participation	ordinal	Sttoolskil	staff tool skills	ordinal
Devenv	development environment adequacy	ordinal	Stteamskil	staff team skills	ordinal
Staffav	staff availability	ordinal	Size	application size	numeric
Standards	standards use	ordinal	Time	start year	numeric
Methods	methods use	ordinal	Effort	effort	numeric
Tools	tools use	ordinal			

Table 5
List of Variables in QQDefects Dataset

Symbol	*Name*	*Type*	*Symbol*	*Name*	*Type*
specexp	Relevant experience of spec & doc staff	ordinal	stexpi	Staff experience - independent test	ordinal
qudoc	Quality of documentation inspected	ordinal	qutestc	Quality of documented test cases	ordinal
regsprev	Regularity of spec & doc reviews	ordinal	devtrq	Dev. staff training quality	ordinal
stproc	Standard procedures followed	ordinal	coman	Configuration management	ordinal
reveff	Review process effectiveness	ordinal	pplan	Project planning	ordinal
specdef	Spec defects discovered in review	ordinal	scco	Scale of distributed communication	ordinal
reqst	Requirements stability	ordinal	stai	Stakeholder involvement	ordinal
compl	Complexity of new functionality	ordinal	cusi	Customer involvement	ordinal
scfunc	Scale of new functionality implemented	ordinal	vman	Vendor management	ordinal
numio	Total no. of inputs and outputs	ordinal	vend	Vendors	nominal
devexp	Relevant development staff experience	ordinal	icomm	Internal communication / interaction	ordinal
procap	Programmer capability	ordinal	pmat	Process maturity	ordinal
defpro	Defined processes followed	ordinal	ptype	Project type	nominal
devstmo	Development staff motivation	ordinal	kloc	project size in KLoC	numeric
tprodef	Testing process well defined	ordinal	effort	effort	numeric
stexpu	Staff experience - unit test	ordinal			

generally: more than a one dataset. Even in such rare cases, some of these variables, for example project size and effort, are typically expressed on different scales. These differences may cause that machine learning techniques may achieve different levels of accuracy depending on the dataset, i.e. a particular technique may perform better in one dataset but worse in another. We examine this when discussing the results in Section 4.2.

2.2. Research Procedure

As indicated in the Section 1, the main assumption for this study is related with applying a research procedure that is simple and easy to follow. Thus, this procedure consists only of very few basic steps.

In the first step an expert has to review the variables in each dataset and keep only these which may be potential predictors for effort, i.e. which are typically known at the stage of the project when effort needs to be predicted. Thus, variables such as "duration" and "number of defects" have been removed because their values are not known in advance.

The second step covers data transformation to new scales, i.e. discretisation. This step is necessary because most techniques selected for use in this experiment work only with categorized data. There are various methods that perform this task automatically, for example to achieve equal-frequency or equal-width intervals. However, such algorithms often define intervals with their starting and ending values that are "not friendly", i.e. not rounded to any significant number. Thus, in this study an expert has performed a discretization for all numeric variables with the aim to define intervals with meaningful values and similar frequencies.

Some machine learning techniques cannot be used with datasets that contain missing data. In our study two datasets, Desharnais and QQDefects, contain missing data. In such cases such missing values are often filled by mean value for numeric variables and the mode for nominal variables. However, to eliminate the bias related with such approach, we have decided to use only those techniques that can work with missing data.

In the main step, each selected technique has been used to generate predictions for each dataset. In this analysis, we have used 23 techniques, listed in Table 6, implemented in a popular Weka tool [8]. We have applied 10-fold cross-validation to generate the predictions. Each dataset has been randomly divided into ten subsets with similar number of cases. Then, subset number 1 has been used as a model/technique testing sample – all remaining nine subsets have been used to build a predictive model and/or generate predictions for each case in the testing sample. This procedure has been repeated for each subset.

Table 6
Machine Learning Techniques used in this Study

Group	*Symbol*	*Technique*
bayes	AODE	Averaged One-Dependence Estimators
	BN	Bayesian Net
	NB	Naïve Bayes
functions	MLP	Multilayer Perceptron
	RBFN	Radial Basis Function Network
	SMO	Sequential Minimal Optimization for Support Vector Machines
lazy	KNN	K-Nearest Neighbour
	K*	K*
	LBR	Lazy Bayesian Rules Classifier
	LWL	Locally Weighted Learning
rules	DT	Decision Table
	DTNB	Decision Table/Naïve Bayes Hybrid
	RIPPER	Repeated Incremental Pruning to Produce Error Reduction
	NNGe	Nearest Neighbor with Generalization
	1R	1R
	PART	PART Decision List
trees	BFT	Best-First Decision Tree
	FT	Functional Trees
	J48	J48
	LADT	Multi-class Alternating Decision Tree
	LMT	Logistic Model Trees
	NBT	Decision Tree with Naïve Bayes Classifiers at the leaves
	RandF	Forest of Random Trees

This step has been performed twice. In the first run, all available predictors have been used to generate a predictive model and provide predictions. In the second run, some predictors in each dataset have been removed using a feature selection technique. Various earlier studies show that more accurate predictions can be achieved not with all available predictors but with only those which are most relevant. In this study we have used a BestFirst feature selection technique implemented in Weka.

In most studies on effort prediction, the assessment of predictive accuracy is based on absolute or relative error, where a predicted numeric value is compared with actual numeric value. However, machine learning techniques used in this study do not predict exact numeric value but one of the classes (intervals). To assess the accuracy such classification task different measures need to be used. These measures assess how accurately, i.e. "how often" a predicted class is the actual class. In this study we have used the following measures:

– accuracy (ACC), a proportion of correctly predicted cases,
– precision (PRE), also called positive predictive value,
– recall (REC), also called sensitivity or true positive rate,
– area under ROC curve (AUC).

In some cases we have also additionally analyzed confusion matrices which summarize the number of correctly and incorrectly predictions for each interval. The aim of this analysis was to examine how often the predicted intervals have lain far from the actual intervals. Other authors have previously used most datasets from this study. Thus, at the end, we have compared the levels of accuracy achieved by these authors.

3. DATA PREPARATION

To satisfy the condition of this experiment that achieving predictions requires very little support from an expert, the data preparation process is also quite simple. One of the key tasks involved here is to discretize the numeric variables into a number of intervals (classes).

Figure 1 illustrates the histograms for "effort" plotted with relatively high number of intervals. Such intervals could not be used in our analysis because in some of them there are very few or zero cases. There is a threat that machine learning techniques would not be able to properly discover relationships between variables using the intervals with such low frequencies. Thus, we need to discretize variables to lower number of intervals, as is common in other studies [9]. In this study, numeric variables have been discretized into four or five intervals depending on a dataset and a particular variable.

An expert has attempted to define a set of intervals with meaningful values and similar frequencies. Table 7 presents such discretization for "effort". It can be observed that these intervals are not of exactly equal frequencies but they can be easily interpreted.

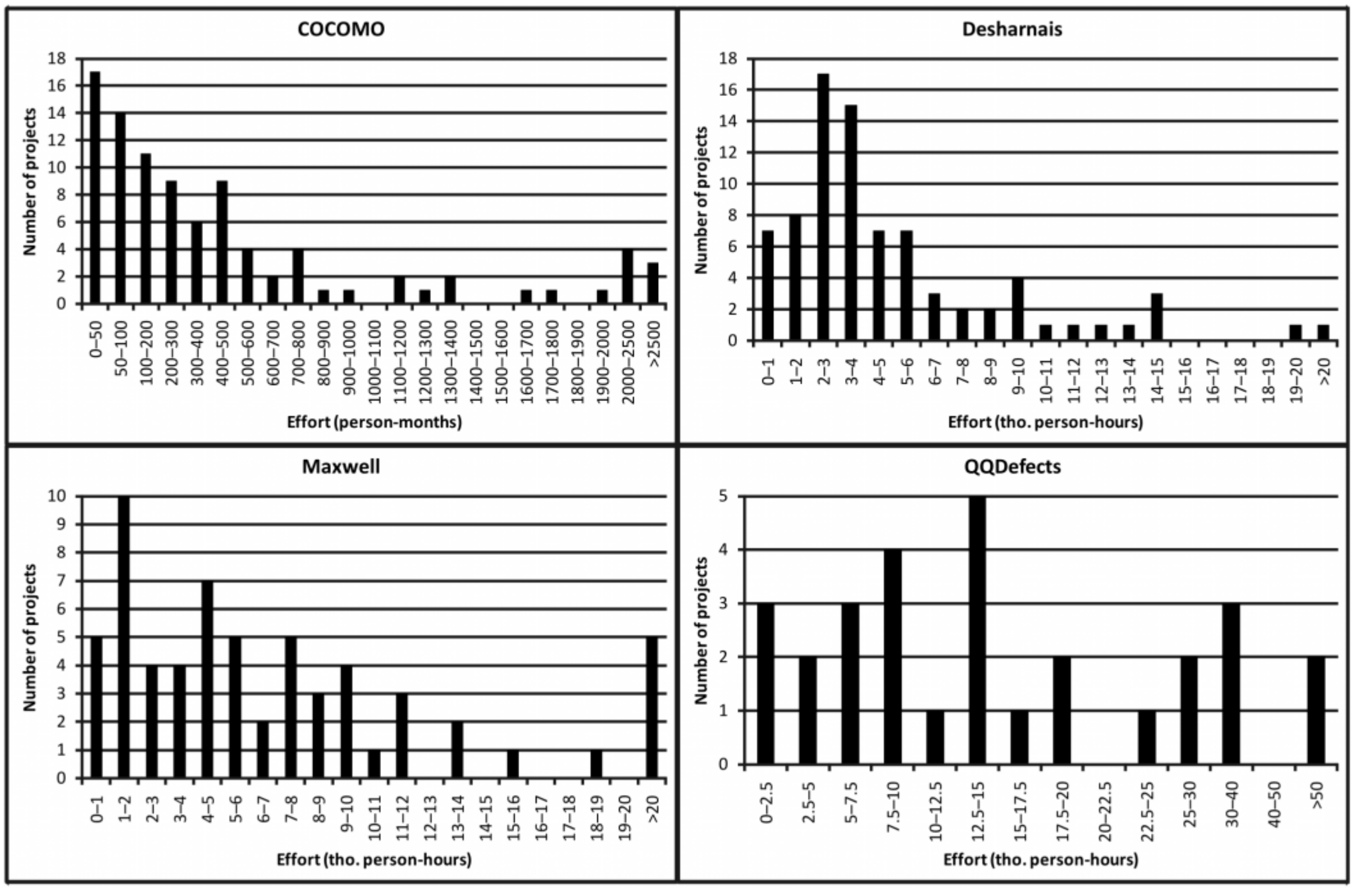

Figure 1: Histograms for Effort in Each Dataset

Table 7
Defined Intervals for Effort in Various Datasets
(Number of Cases in Brackets)

Dataset / *Class ID*	*COCOMO*		*Desharnais*		*Maxwell*		*QQDefects*	
1	0 – 50	(17)	0 – 1500	(11)	0 – 1500	(9)	0 – 5000	(5)
2	50 – 120	(19)	1500 – 3000	(21)	1500 – 3000	(10)	5000 – 10000	(7)
3	120 – 400	(20)	3000 – 4500	(20)	3000 – 5000	(11)	10000 – 20000	(9)
4	400 – 1000	(22)	4500 – 8000	(14)	5000 – 10000	(18)	> 20000	(8)
5	> 1000	(15)	> 8000	(15)	> 10000	(14)		

In this experiment, the predictions have been achieved in two modes – with and without feature selection. In the second mode, we have used the BestFirst method to select relevant features (i.e. predictors). Table 8 contains lists of nominated predictors used in this mode for each dataset. As expected, a variable reflecting project size has been selected in all four datasets. Surprisingly, for COCOMO dataset, only two other predictors have been selected, and none of them describes the development process.

Table 8
Predictors Selected in Feature Selection Stage

Predictors \ *Dataset*	*COCOMO*	*Desharnais*	*Maxwell*	*QQDefects*
Numeric	kloc	Transactions Entities FPNonAdjust Adjustment	Nlan Size	kloc
Ranked	–	–	Custpart Staffav Complex Instreq	reqst compl defpro scco pmat
Nominal	projectname mode	Language	Dba	–

4. RESULTS

4.1. Basic data analysis

Prior to assessing and discussing the accuracy of predictions we have performed a basic data analysis to better understand the data gathered in analyzed datasets. The focus of this basic analysis is on identifying most important predictors and on determining the strength of the relationship between effort and each numeric predictor. This step does not necessarily need to be followed when assessing the accuracy of machine learning techniques.

Correlation coefficients are popular and easy in interpretation measures of the strength of relationships between numeric variables. Figure 2 illustrates the values of Spearman rank correlation coefficient (SRCC) between effort and each numeric

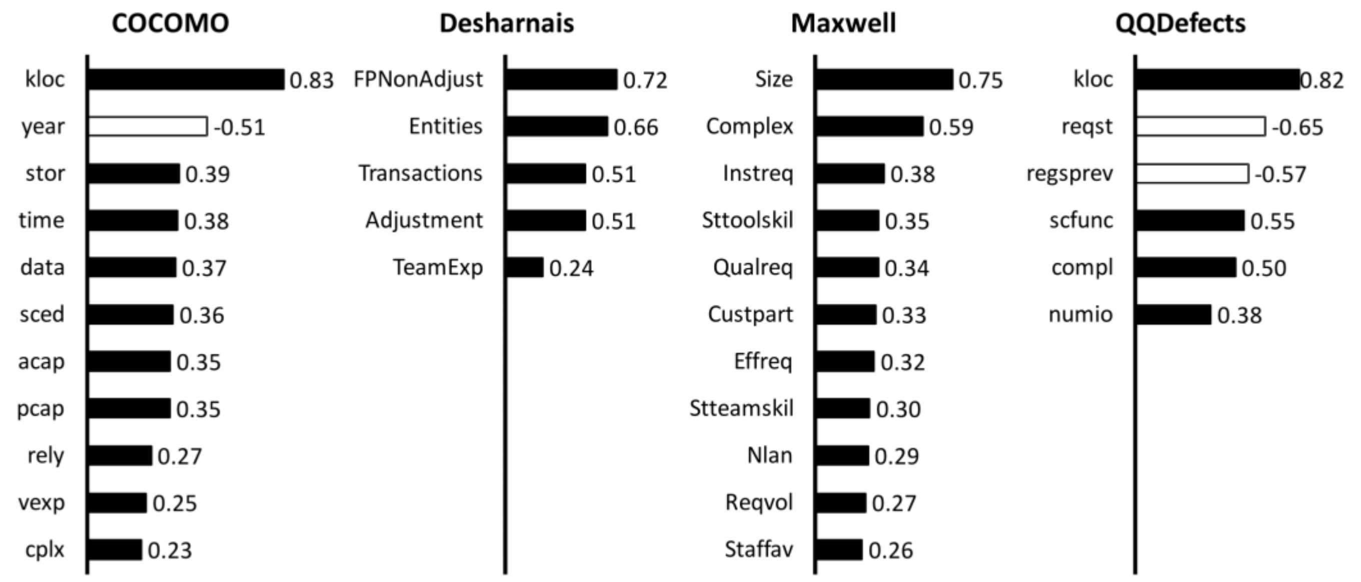

Figure 2: Values of Spearman Rank Correlation Coefficients between Effort and Relevant Predictors

predictor. Only those variables have been illustrated, for which the value of SRCC has been statistically significant at $p<0.05$. Dark bars indicate the positive values of SRCC and white bars – negative values of SRCC.

Project development effort has the strongest correlation with the size of the project – this can be observed in all analyzed datasets. However, the strength of impact of other factors varies among the datasets. For example, complexity is the second factor strongly related with effort in Maxwell dataset (SRCC=0.59), fifth in QQDefects (SRCC=0.50) and only eleventh in COCOMO (SRCC=0.23).

Figure 3 provides a visualization, in the form of scatterplot, of the relationships between project size and development effort for each dataset. We can see that with an increase of project size the development effort also increases. However, a high variability in this relationship indicates that there are other factors significantly influencing development effort.

4.2. Accuracy of Predictions

The values of measures of predictive accuracy for a run with no feature selection are summarized in Tables 9-10. These values fall in the range <0, 1>. The value closer to one indicates a higher predictive accuracy. For each dataset and each accuracy measure the top three best performing techniques have been marked with underline, and three worst performing techniques with strikethrough.

It can be observed that low values of these measures, even for best performing techniques, suggest very low accuracy of predictions. The highest accuracy have been achieved with COCOMO dataset using LADT technique. Both ACC, PRE and REC reached a value of 0.68, and AUC of 0.87. No other technique provided higher values for ACC, PRE, REC and AUC in any dataset. The lowest overall accuracy has been achieved using K* technique with Maxwell dataset, where both ACC, PRE and REC reached a value of 0.13, and AUC of 0.48. However, even lower values for

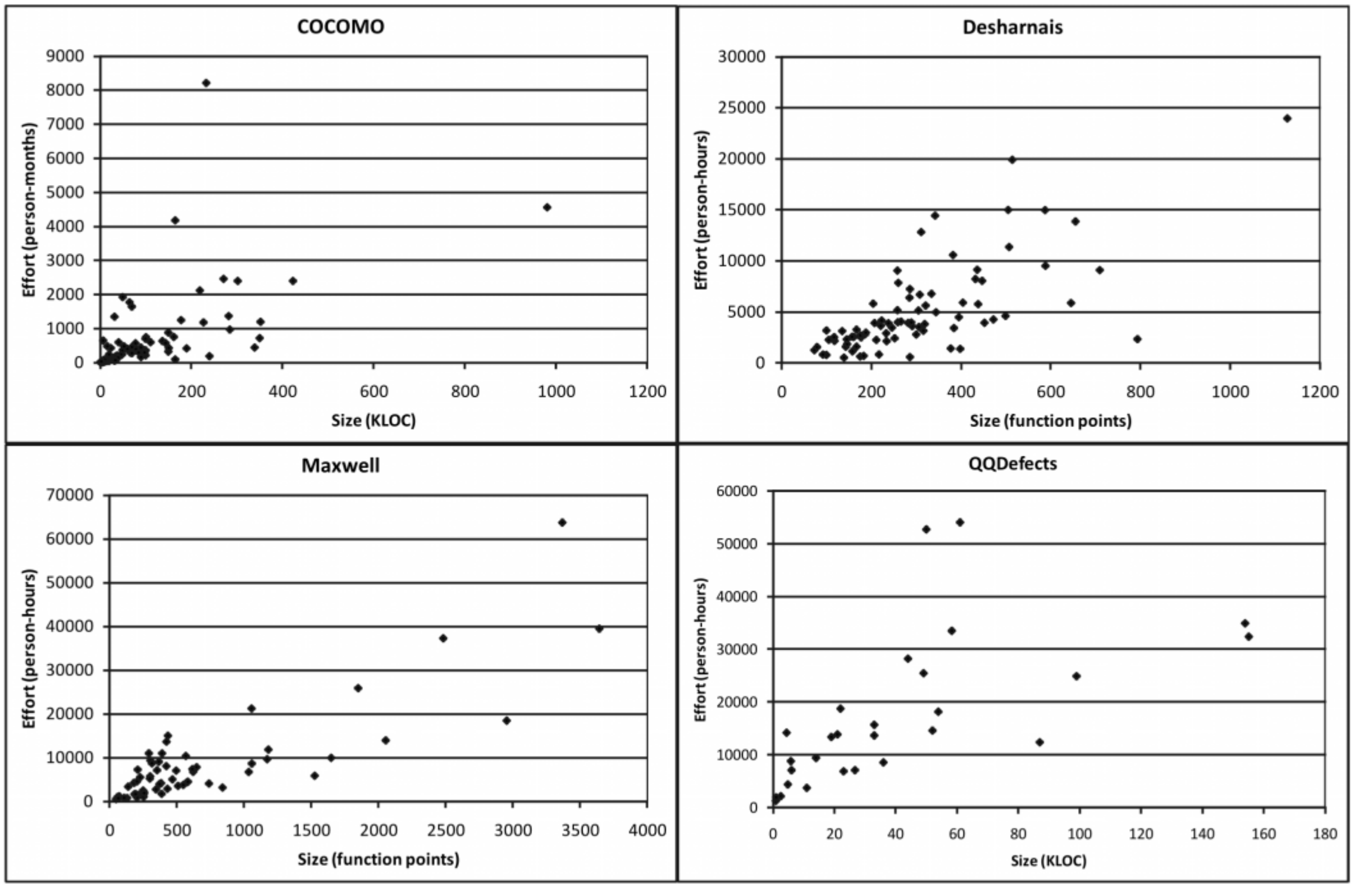

Figure 3: Scatterplots for Project Size vs Effort

selected measures have been observed, for example in QQDefects PRE=0.10 using FT technique, while AUC=0.43 using LWL and AUC=0.44 using NBT.

The accuracy of predictions for almost all techniques was not stable across different datasets. For example, while LADT produced most accurate predictions for COCOMO, it has been significantly less accurate with Desharnais dataset and very inaccurate with Maxwell and QQDefects datasets. This lack of stability has been observed not only for most accurate techniques but also for the least accurate. Techniques least accurate in one dataset often performed better than other techniques with different datasets. The only two exceptions could be K* and KNN techniques which have consistently provided one of the least accurate predictions for both Maxwell and QQDefects. Still, they are ranked in the middle among other techniques with COCOMO and Desharnais datasets. This lack of stability of predictive accuracy causes a problem in recommending a particular technique for industrial use.

Tables 11-12 summarize the values of predictive accuracy for the same techniques in the run with feature selection. Generally, these values are higher than in the run without feature selection (Tables 9-10), what suggests higher accuracy achieved by these techniques using feature selection. Interestingly, some techniques that performed very poor without feature selection have now achieved predictions more accurate than other techniques. For example, while K* has performed very poor without feature selection, it has been one of the best performing in COCOMO, Desharnais and

Table 9
Measures of Predictive Accuracy – no Feature Selection, COCOMO and Desharnais Datasets

Dataset and measure	*COCOMO*				*Desharnais*			
Technique	*ACC*	*PRE*	*REC*	*AUC*	*ACC*	*PRE*	*REC*	*AUC*
AODE	0.43	~~0.38~~	0.43	0.74	0.44	0.42	0.44	**0.78**
BN	~~0.39~~	~~0.37~~	~~0.39~~	0.76	0.48	0.46	0.48	0.78
NB	~~0.40~~	~~0.38~~	~~0.40~~	0.75	0.46	0.44	0.46	**0.78**
MLP	0.51	0.51	0.51	0.77	0.41	0.41	0.41	0.73
RBFN	~~0.39~~	~~0.38~~	~~0.39~~	~~0.68~~	0.44	0.44	0.44	0.73
SMO	0.53	0.55	0.53	0.79	0.47	0.47	0.47	0.75
KNN	0.47	0.48	0.47	0.73	0.41	0.39	0.41	~~0.65~~
K*	0.46	0.46	0.46	0.78	0.43	0.39	0.43	0.76
LBR	0.41	0.40	0.41	0.76	0.48	0.46	0.48	**0.78**
LWL	0.54	0.56	0.54	0.82	0.47	0.46	0.47	0.72
DT	0.41	0.42	0.41	0.75	~~0.37~~	~~0.36~~	~~0.37~~	0.68
DTNB	0.57	0.57	0.57	**0.84**	**0.51**	**0.51**	**0.51**	**0.78**
RIPPER	**0.60**	0.59	**0.60**	0.81	0.47	0.45	0.47	0.68
NNGe	0.52	0.51	0.52	~~0.69~~	0.46	0.43	0.46	**0.66**
1R	0.56	0.56	0.56	~~0.72~~	~~0.37~~	~~0.30~~	~~0.37~~	~~0.59~~
PART	**0.68**	**0.68**	**0.68**	0.83	0.44	0.44	0.44	0.73
BFT	0.46	0.47	0.46	0.76	**0.51**	**0.53**	**0.51**	0.68
FT	0.53	0.53	0.53	0.78	0.47	0.46	0.47	0.67
J48	**0.60**	**0.60**	**0.60**	0.80	~~0.38~~	0.39	~~0.38~~	0.68
LADT	**0.68**	**0.68**	**0.68**	**0.87**	0.44	0.46	0.44	0.71
LMT	0.57	0.57	0.57	0.82	**0.56**	**0.56**	**0.56**	0.76
NBT	0.59	0.59	0.59	**0.85**	~~0.38~~	~~0.38~~	~~0.38~~	0.73
RandF	0.48	0.49	0.48	0.83	0.44	0.43	0.44	0.78

Table 10
Measures of Predictive Accuracy – No Feature Selection, Maxwell and QQDefects Datasets

Dataset and measure	*Maxwell*				*QQDefects*			
Technique	*ACC*	*PRE*	*REC*	*AUC*	*ACC*	*PRE*	*REC*	*AUC*
AODE	0.24	0.23	0.24	0.56	0.38	0.39	0.38	0.61
BN	0.31	**0.33**	0.31	0.62	**0.48**	**0.49**	**0.48**	**0.67**
NB	0.31	**0.33**	0.31	0.62	**0.41**	0.40	**0.41**	0.64
MLP	0.29	0.29	0.29	0.60	0.34	0.36	0.35	**0.66**
RBFN	0.29	0.32	0.29	0.59	**0.41**	0.41	**0.41**	0.60
SMO	0.26	0.30	0.26	**0.64**	0.31	0.33	0.31	0.56
KNN	~~0.18~~	0.19	~~0.18~~	~~0.49~~	~~0.24~~	0.24	~~0.24~~	~~0.48~~
K*	~~0.13~~	~~0.13~~	~~0.13~~	~~0.48~~	~~0.24~~	~~0.23~~	~~0.24~~	0.56
LBR	0.31	0.32	0.31	0.61	**0.48**	**0.47**	**0.48**	0.65
LWL	0.29	~~0.18~~	0.29	0.56	~~0.17~~	~~0.16~~	~~0.17~~	~~0.43~~
DT	0.24	0.28	0.24	0.61	0.28	0.41	0.28	0.55
DTNB	**0.35**	0.32	**0.36**	**0.62**	0.34	0.35	0.35	0.56
RIPPER	**0.39**	**0.39**	**0.39**	0.60	0.28	0.30	0.28	0.51
NNGe	0.31	0.29	0.31	0.55	**0.41**	**0.44**	**0.41**	0.59
1R	**0.35**	0.31	**0.36**	0.59	0.34	0.35	0.35	0.57
PART	~~0.19~~	~~0.18~~	~~0.19~~	~~0.48~~	0.28	0.29	0.28	0.55
BFT	0.27	0.32	0.27	0.56	0.38	0.27	0.38	0.60
FT	0.29	0.32	0.29	0.60	0.31	~~0.10~~	0.31	0.50
J48	0.27	0.26	0.27	0.57	0.38	0.35	0.38	0.58
LADT	0.27	0.26	0.27	0.57	0.28	0.27	0.28	0.49
LMT	0.27	0.33	0.27	0.58	~~0.24~~	0.30	~~0.24~~	~~0.44~~
NBT	0.31	0.32	0.31	0.55	**0.41**	0.40	**0.41**	0.64
RandF	0.29	0.25	0.29	**0.62**	0.31	0.34	0.31	**0.66**

QQDefects datasets and not very far from the top with Maxwell dataset. The LADT technique, which achieved the highest accuracy without feature selections, has been one of the least accurate with Maxwell dataset and around the middle among other techniques in other datasets. Such results confirm that feature selection can significantly change the accuracy of predictions.

What is most important is that the stability of predictive accuracy has increased. It means that very often a particular technique, which has performed well in one dataset, it has also performed well with other datasets.

Table 11
Measures of Predictive Accuracy – with Feature Selection, COCOMO and Desharnais Datasets

Dataset and measure	*COCOMO*				*Desharnais*			
Technique	*ACC*	*PRE*	*REC*	*AUC*	*ACC*	*PRE*	*REC*	*AUC*
AODE	0.62	0.63	0.62	**0.88**	0.53	0.53	0.53	**0.82**
BN	0.56	0.57	0.56	0.85	0.54	0.54	0.54	0.82
NB	0.57	0.58	0.57	0.85	**0.56**	**0.56**	**0.56**	**0.82**
MLP	**0.68**	**0.68**	**0.68**	0.86	0.51	0.51	0.51	0.78
RBFN	0.67	0.67	0.67	0.84	0.54	0.55	0.54	0.73
SMO	0.67	**0.68**	0.67	0.88	**0.56**	**0.59**	**0.56**	0.80
KNN	**0.68**	0.68	**0.68**	0.86	0.48	0.50	0.48	0.75
K*	**0.69**	**0.69**	**0.69**	**0.89**	**0.56**	0.56	**0.56**	0.80
LBR	0.58	0.59	0.58	0.85	**0.56**	**0.56**	**0.56**	**0.82**
LWL	0.57	0.57	0.57	0.84	0.53	0.54	0.53	0.72
DT	~~0.53~~	~~0.53~~	~~0.53~~	0.78	~~0.37~~	~~0.36~~	~~0.37~~	~~0.68~~
DTNB	~~0.51~~	~~0.52~~	~~0.51~~	0.83	0.52	0.56	0.52	0.78
RIPPER	~~0.48~~	~~0.52~~	~~0.48~~	~~0.77~~	~~0.41~~	~~0.41~~	~~0.41~~	~~0.69~~
NNGe	0.57	0.58	0.57	~~0.73~~	**0.56**	**0.57**	**0.56**	0.72
1R	0.56	0.56	0.56	~~0.72~~	~~0.37~~	~~0.30~~	~~0.37~~	~~0.59~~
PART	0.58	0.58	0.58	0.82	0.51	0.52	0.51	0.72
BFT	0.62	0.63	0.62	0.82	0.52	0.55	0.52	0.74
FT	0.61	0.62	0.61	0.82	0.54	0.55	0.54	0.76
J48	0.58	0.59	0.58	0.81	0.48	0.50	0.48	0.71
LADT	0.62	0.64	0.62	**0.88**	0.46	0.45	0.46	0.77
LMT	0.56	0.57	0.56	0.81	0.51	0.51	0.51	0.72
NBT	0.57	0.58	0.57	0.85	0.53	0.54	0.53	0.78
RandF	0.65	0.65	0.65	0.83	0.48	0.48	0.48	0.79

Table 12
Measures of Predictive Accuracy – with Feature Selection, Maxwell and QQDefects Datasets

Dataset and measure	*Maxwell*				*QQDefects*			
Technique	*ACC*	*PRE*	*REC*	*AUC*	*ACC*	*PRE*	*REC*	*AUC*
AODE	**0.44**	0.44	**0.44**	0.71	**0.52**	0.53	**0.52**	**0.82**
BN	0.42	**0.46**	0.42	**0.75**	**0.55**	**0.56**	**0.55**	**0.82**
NB	0.42	**0.46**	0.42	**0.75**	**0.52**	0.54	**0.52**	0.82
MLP	0.40	0.42	0.40	0.70	0.45	0.46	0.45	0.72
RBFN	**0.45**	0.44	**0.45**	0.67	**0.55**	**0.56**	**0.55**	0.69
SMO	0.37	0.39	0.37	0.68	0.48	**0.54**	0.48	0.75
KNN	0.32	0.33	0.32	0.66	**0.52**	0.51	**0.52**	0.69
K*	0.42	0.42	0.42	0.69	**0.52**	0.51	**0.52**	0.77
LBR	0.42	**0.46**	0.42	**0.75**	**0.52**	0.54	**0.52**	**0.83**
LWL	0.32	~~0.27~~	0.32	0.63	~~0.34~~	0.36	~~0.35~~	~~0.49~~
DT	~~0.31~~	0.33	~~0.31~~	0.66	~~0.24~~	~~0.31~~	~~0.24~~	0.60
DTNB	0.39	0.38	0.39	0.63	0.38	0.42	0.38	0.70
RIPPER	0.42	0.43	0.42	0.62	0.45	0.51	0.45	0.67
NNGe	0.40	0.43	0.40	0.62	0.45	0.49	0.45	0.63
1R	0.35	0.31	0.36	~~0.59~~	~~0.34~~	0.35	~~0.35~~	~~0.57~~
PART	~~0.31~~	~~0.31~~	~~0.31~~	0.65	0.38	0.37	0.38	0.64
BFT	~~0.29~~	~~0.30~~	~~0.29~~	~~0.58~~	~~0.34~~	~~0.30~~	~~0.35~~	0.64
FT	0.35	0.35	0.36	~~0.59~~	~~0.31~~	~~0.10~~	~~0.31~~	~~0.50~~
J48	0.34	0.33	0.34	0.62	0.45	0.33	0.45	0.61
LADT	~~0.29~~	0.32	~~0.29~~	0.64	0.38	0.40	0.38	0.62
LMT	**0.44**	**0.47**	**0.44**	0.69	0.38	0.41	0.38	0.68
NBT	0.35	0.38	0.36	0.65	**0.52**	0.54	**0.52**	0.82
RandF	0.35	0.36	0.36	0.67	0.45	0.44	0.45	0.74

Table 13 illustrates the impact of using feature selection on predictive accuracy. These values indicate the difference of values of accuracy measures between modes with using and not using feature selection. The highest gain from feature selection has been observed for K* technique on Maxwell dataset with delta ACC=0.29 and delta AUC=0.21. The impact of feature selection on ACC, PRE and REC was in most cases very similar, thus, for clarity we publish only the delta for ACC.

KNN is another technique that has significantly improved on all datasets. Other techniques have improved significantly only on selected datasets. For example, RBFN has achieved higher accuracy only on COCOMO, and much less significantly on other datasets. There are also some techniques,

Table 13
The Impact of Feature Selection on the Predictive Accuracy

Dataset and measure	*COCOMO*		*Desharnais*		*Maxwell*		*QQDefects*	
Technique	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*
AODE	0.19	**0.14**	0.09	0.04	**0.19**	0.16	0.14	**0.22**
BN	0.17	0.09	0.06	0.04	0.11	0.14	0.07	0.15
NB	0.17	0.10	**0.10**	0.04	0.11	0.14	0.10	0.18
MLP	0.17	0.08	**0.10**	0.05	0.11	0.10	0.10	0.06
RBFN	**0.28**	**0.16**	0.10	0.00	0.16	0.08	0.14	0.09
SMO	0.14	0.08	0.09	0.05	0.11	0.05	**0.17**	0.19
KNN	**0.20**	**0.13**	0.07	**0.10**	0.15	**0.17**	**0.28**	**0.21**
K*	**0.23**	0.11	**0.12**	0.04	**0.29**	**0.21**	**0.28**	0.21
LBR	0.17	0.09	0.07	0.04	0.11	0.14	0.03	0.18
LWL	0.03	0.03	0.06	0.00	0.03	0.06	**0.17**	0.06
DT	0.12	0.03	~~0.00~~	0.00	0.06	0.05	~~-0.03~~	0.05
DTNB	~~-0.06~~	~~-0.01~~	0.01	~~0.00~~	0.03	~~0.01~~	0.03	0.14
RIPPER	~~-0.12~~	~~-0.04~~	~~-0.06~~	0.01	0.03	0.02	**0.17**	0.16
NNGe	0.05	0.04	**0.10**	**0.07**	0.10	0.07	0.03	0.03
1R	0.00	0.00	~~0.00~~	0.00	~~0.00~~	~~0.00~~	~~0.00~~	~~0.00~~
PART	~~-0.10~~	0.00	0.06	~~-0.01~~	0.11	**0.16**	0.10	0.09
BFT	0.16	0.06	0.01	0.06	~~0.02~~	0.02	~~-0.03~~	0.04
FT	0.09	0.05	0.07	**0.09**	0.06	~~-0.01~~	~~0.00~~	~~0.00~~
J48	-0.02	0.01	0.10	0.03	0.06	0.04	0.07	~~0.02~~
LADT	-0.05	0.01	0.01	0.06	~~0.02~~	0.07	0.10	0.13
LMT	-0.01	0.00	~~-0.05~~	~~-0.04~~	**0.16**	0.11	0.14	**0.24**
NBT	-0.02	~~-0.01~~	**0.15**	0.05	0.05	0.11	0.10	0.18
RandF	0.16	0.00	0.04	0.02	0.06	0.05	0.14	0.08

which have not improved at all with feature selection, and some have even achieved a lower accuracy. Among them, there are for example DTNB, RIPPER, 1R, BFT, FT, and LMT on selected datasets.

Figure 4 illustrates predictions, i.e. confusion matrices, for K* technique with and without feature selection for COCOMO dataset. The values on the diagonal, that indicate the number of correctly predicted intervals, have been marked with bold frame. The K* technique with feature selection for COCOMO dataset provides the highest accuracy among all other techniques and all datasets. We can observe that in most cases the interval for development effort has been correctly predicted. For incorrectly predicted cases the majority achieved the prediction within the neighboring interval, only two cases in a distance of two intervals from the actual values, and another two cases three intervals from original predictions. Predictive accuracy expressed using various measures is far from perfect even for the best performing technique. However, such detailed results show that for incorrectly predicted cases the difference between actual and predicted values was not high.

Without using feature selection the predictions achieved by K* on COCOMO dataset have been significantly worse, both in terms of accurately predicted cases and in term of the difference between actual and predicted values. However, even with ACC=0.46, only 6/93 cases (6.5%) have been predicted with a difference of more than two intervals from the originals. Thus, such accuracy may still be satisfactory in some environments.

4.3. Related Work

Other authors have previously used the same four datasets in their studies. It is beyond the scope of this paper to discuss all of earlier studies but we bring some of them. For COCOMO dataset Srinivasan and Fisher [19] have found that neural network outperformed regression tree in predictive accuracy. Baskeles *et al.* [1] has achieved the highest accuracy using decision trees, followed by RBF network, support vector machines and MLP.

For Desharnais dataset Li *et al.* [11] have achieved the most accurate predictions with various techniques related to estimation by analogy, followed by stepwise regression, neural networks and regression trees. Schepperd and Schofield [18] have achieved more accurate predictions using estimation by analogy than with stepwise regression model. Mair *et al.* [13] used various techniques for predicting effort with this dataset. They have achieved the highest accuracy with neural networks, followed by case-based reasoning and least squares regression models (*ex aequo*) and rule induction.

Li *et al.* [11] have compared the accuracy of various techniques also for Maxwell dataset. They have achieved the most accurate predictions with various techniques based on estimation by analogy, followed by regression trees, stepwise regression and neural network models. Using the same dataset Malhotra *et al.* [14] have found that linear

without feature selection

		Actual				
		1	**2**	**3**	**4**	**5**
	1	11	7	2	2	2
	2	5	4	4	1	0
Predicted	**3**	0	6	10	3	4
	4	1	1	3	12	3
	5	0	1	1	4	6

with feature selection

		Actual				
		1	**2**	**3**	**4**	**5**
	1	11	1	0	2	0
	2	6	15	2	0	0
Predicted	**3**	0	2	17	3	0
	4	0	1	0	11	5
	5	0	0	1	6	10

Figure 4: Comparison of Predictions using K* Technique without (Left) and with (Right) Feature Selection on COCOMO Dataset

regression, MSP and M5Rules are effective techniques for predicting software development effort; support vector machine and RBF models provided significantly lower accuracy.

QQDefects, as explained earlier, has been originally used for defect prediction [6]. However, in one study it has also been used for effort prediction. Radliński [17] has found that Bayesian net model automatically generated from data has not provided accurate predictions. To improve them a stronger input from domain expert would be required.

In all these related studies the authors have used different measures of accuracy – based on relative or absolute error, not on accuracy of classification (see discussion in the next section). We have found only one study where the authors have used similar accuracy measures as in this paper. Hewett and Kijsanayothin [9] have used various machine learning techniques for predicting defect repair time using a dataset from an earlier study [20]. They have achieved the highest accuracy with J48 and DT. In our study, J48 has achieved high accuracy only for COCOMO dataset with no feature selection, in other cases the accuracy was among the middle. The DT technique has been consistently among the least accurate techniques for various datasets.

5. THREATS TO VALIDITY

In most cases the accuracy of particular techniques has been unstable across datasets. There are significant differences between these datasets, in number of cases and in number of variables of particular type. These differences in the datasets may have an impact on the accuracy of predictions. We plan to investigate this problem in further analyses.

This experiment involves using only local data, i.e. from a single company. We have found only four publicly available datasets that meet this condition. Further, for this reason analyzed datasets are relatively small because a single company does not develop as many projects as a set of companies. Therefore, based on the achieved results, we believe that it is difficult to generalize how particular machine learning technique perform in effort prediction.

With low number of cases in the datasets, in particular in QQDefects, we have decided to discretize the numeric values to relatively low number of intervals (states). It has caused an inevitable loss of precision of expressing the values of these variables. However, with higher number of intervals there is a risk that machine learning techniques would not be able to properly learn "the rules" from the datasets.

Development effort is a number expressed in person-hours or person-months; other studies may use some other units. However, in this study the aim is not to predict the numeric value for effort but a correct interval. Thus, the accuracy of predictions have been expressed with measures such as ACC, PRE, REC, and AUC which are typically used in classification tasks. These measures are based on a concept that correct prediction occurs when predicted interval is the same as actual interval and incorrect prediction occurs when predicted interval is different than actual interval. These measures do not contain information of the degree of inaccuracy, i.e. how far the incorrect prediction is from the actual interval. Other studies focused on effort prediction use measures based of relative or absolute error calculated for numeric values of effort. Thus, a comparison of the accuracy achieved in this study with the accuracy achieved in other studies cannot be done directly.

In industry development effort is predicted for specific projects. These projects are not anonymous and can be described using a set of nominal features. The datasets used in this study contain very few such factors, especially in the run with feature selection. Thus, not all data/knowledge typically available in industrial setting has been used in predictions.

Several techniques can be used with various values of their parameters. In this experiment we have not analyzed the impact of the values of these parameters. For example, in KNN it is necessary to provide a value for "k" – the number of neighbors. In this study we have used a value "1", but in other studies also other values have been considered, for example "3" and "5" in [9]. The values of such parameters may influence the accuracy of predictions. In future we plan to investigate the impact of these parameters for some techniques.

We have shown in the paper that the most influential factor for development effort is project size, typically expressed in function points or KLOC. There is a problem with using project size as predictor for effort. Effort often needs to be predicted at the beginning of the project. In such cases precise value of project size is often unknown. This happens especially when it is measured in KLOC – the code is not finished, thus the release version cannot be measured. Expressing project size in function points may be performed earlier, when detailed software design is ready. However, in some development approaches, for example agile, such design is not prepared early in project lifecycle. As pointed earlier, all predictions in this study have been performed not using the numeric values of project size but using the discretized intervals. In reality such discretisation may correspond to assessment of project size on a four- or five-point ranked scale, which is easier to achieve than estimating a numeric value.

6. SUMMARY

In this study we have compared the accuracy of predictions for software development effort provided by 23 machine learning techniques across four datasets. The key idea of this study is to perform an experiment under two constraints: using only local data and applying analysis procedure that is easy to follow.

In the first run, we have used data for all available variables. The accuracy of particular techniques is different depending on the dataset used. The differences between the most and the least accurate techniques are high.

In the second run, we repeated this experiment on datasets reduced after applying a feature selection technique. We have observed higher predictive accuracy for majority of techniques. However, some techniques provided least accurate predictions than in the first run. The K* technique appear to be the most accurate. It is also among the techniques that have achieved the highest improvement on accuracy with feature selection.

The general level of accuracy, expressed with various measures for classification tasks, is not very high. Such results suggest that predicted effort intervals have been different from actual intervals. However, in the detailed analysis of confusion matrices we have revealed that very often the distance between the predicted and actual interval have been low. Thus, such techniques may not provide very precise predictions in such easy-to-follow procedure, but the level of inaccuracy may still be acceptable for project managers who can fast get a general overview of the development effort.

In future, we plan to further investigate the applicability of such easy procedure. Specifically, we plan to analyze issues raised in Section 5, related to the features of the datasets and the projects collected in them, as well as using various parameters of machine learning techniques. Such analysis can also be extended for other areas of software engineering such as defect prediction.

References

[1] Baskeles B., Turhan B., Bener A., Software Effort Estimation Using Machine Learning Methods, 22nd International Symposium on Computer and Information Sciences, Ankara, 2007, pp. 1-6.

[2] Boehm B. W., Abts C., Brown W., Chulani S., Clark B. K., Horowitz E., Madachy R., Reifer D. J., Software Cost Estimation with Cocomo II, Englewood Cliffs, NJ: Prentice Hall PTR, 2000.

[3] Boehm B. W., Software Engineering Economics, Upper Saddle River, NJ: Prentice Hall PTR, 1981.

[4] Boetticher G., Menzies T., Ostrand T., PROMISE Repository of empirical software engineering data, West Virginia University, Department of Computer Science, 2010, *http://promisedata.org/repository.*

[5] Desharnais J. M., Analyse Statistique de la Productivitie des projets informatique a partie de la technique des point des function, Master Thesis, University of Montreal, 1989.

[6] Fenton N., Neil M., Marsh W., Hearty P., Radliński Ł., Krause P., On the Effectiveness of Early Life Cycle Defect Prediction with Bayesian Nets, *Empirical Software Engineering*, **13**(5), 2008, 499-537.

[7] Fenton N., Neil M., Marsh W., Hearty P., Radliński Ł., Krause P., Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction, Proceedings 3rd International Workshop on Predictor Models in Software Engineering. International Conference on Software Engineering, Washington, DC: IEEE Computer Society, 2007, p. 2.

[8] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H., The WEKA Data Mining Software: An Update, SIGKDD Explorations, **11**(1), 2009.

[9] Hewett R., Kijsanayothin P., On Modeling Software Defect Repair Time, *Empirical Software Engineering*, **14**(2), 2009, 165-186.

[10] Jørgensen M., Shepperd M., A Systematic Review of Software Development Cost Estimation Studies, *IEEE Transactions on Software Engineering*, **33**(1), 2007, 33-53.

[11] Li Y. F., Xie M., Goh T. N., A Study of the Non-linear Adjustment for Analogy Based Software Cost Estimation, *Empirical Software Engineering*, **14**(6), 2009, 603-643.

[12] Lokan C., Mendes E., Cross-company and Single-company Effort Models using the ISBSG Database: A Further Replicated Study, Proceedings of the 2006 ACM/IEEE international Symposium on Empirical Software Engineering, New York: ACM, 2006, pp. 75-84.

[13] Mair C., Kadoda G., Lefley M., Phalp K., Schofield C., Shepperd M., Webster S., An Investigation of Machine Learning based Prediction Systems, *Journal of Systems Software*, **53**(1), 2000, 23-29.

[14] Malhotra R., Kaur A., Singh G.G., Application of Machine Learning Methods for Software Effort Prediction, *ACM SIGSOFT Software Engineering Notes*, **35**(3), 2010, 1-6.

[15] Maxwell K. D., Applied Statistics for Software Managers, Upper Saddle River, NJ: Prentice Hall PTR, 2002.

[16] Mendes E., Lokan C. Replicating Studies on Cross- vs Single-company Effort Models using the ISBSG Database, *Empirical Software Engineering*, **13**(1), 2008, 3-37.

[17] Radliński Ł., Software Development Effort and Quality Prediction Using Bayesian Nets and Small Local Qualitative Data, in Proc. 22nd International Conference on Software Engineering and Knowledge Engineering, Redwood City, CA, 2010, pp. 113-116.

[18] Shepperd M. J., Schofield C., Estimating Software Project Effort using Analogies, *IEEE Transactions on Software Engineering*, **23**, 1997, 736-743.

[19] Srinivasan K., Fisher D., Machine Learning Approaches to Estimating Software Development Effort, *IEEE Transactions on Software Engineering*, **21**(2), 1995, 126-137.

[20] Stringfellow C., Andrews A., An Empirical Method for Selecting Software Reliability Growth Models, *Empirical Software Engineering*, **7**(4), 2002, 319–343.