Support Vector Machine Regression for project control forecasting

Mathieu Wauters[a], Mario Vanhoucke[a,b,c,*]

[a]*Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Gent (Belgium)*
[b]*Technology and Operations Management, Vlerick Business School, Reep 1, 9000 Gent (Belgium)*
[c]*Department of Management Science and Innovation, University College London, Gower Street, London WC1E 6BT (United Kingdom)*

Abstract

Support Vector Machines are methods that stem from Artificial Intelligence and attempt to learn the relation between data inputs and one or multiple output values. However, the application of these methods has barely been explored in a project control context. In this paper, a forecasting analysis is presented that compares the proposed Support Vector Regression model with the best performing Earned Value and Earned Schedule methods. The parameters of the SVM are tuned using a cross-validation and grid search procedure, after which a large computational experiment is conducted. The results show that the Support Vector Machine Regression outperforms the currently available forecasting methods. Additionally, a robustness experiment has been set up to investigate the performance of the proposed method when the discrepancy between training and test set becomes larger.

Keywords: project management, earned value management (EVM), Support Vector Regression (SVR), prediction

1. Introduction

Project scheduling first originated as a subdiscipline of Operations Research with the goal of establishing start and finish times of activities within a project network. These activities are subject to various types of constraints, of which precedence and resource restrictions are the most renowned, while optimizing a certain objective. While the construction of a baseline schedule plays a vital role in the ultimate failure or success of a project, its primary purpose consists of acting as a point of reference. The assessment of a project's risk and the analysis of a project's performance throughout its lifecycle are compared against this predictive plan. Dynamic scheduling (Uyttewael (2005), Vanhoucke (2012)) refers to these three crucial phases in a project's life cycle, namely baseline scheduling, schedule risk analysis and project control. Ever since the inception of the well-known

[*]Corresponding author
Email addresses: `mathieu.wauters@ugent.be` (Mathieu Wauters), `mario.vanhoucke@ugent.be` (Mario Vanhoucke)

Critical Path Method in the 1950s, the research community focused heavily on project scheduling problems with various extensions. The PERT methodology turned the attention of academics towards the relation between the duration of a project and variability affecting activity durations. The third component of dynamic scheduling is project control. Earned Value Management (EVM) was introduced as a methodology to control a project's time and cost and aids a project manager in keeping track of the execution of a project vis-à-vis the reference point, provided by the baseline schedule. It surfaced in the 1960s thanks to a project of the US Department of Defense. The reader is referred to Fleming and Koppelman (2005) for the fundamentals of EVM.

A popular project control topic was the search for accurate and reliable forecasting methods. Forecasting methods that provide a project manager with a reliable estimate of the project's targets are an important asset in the project manager's toolbox. Depending on the allowed deviation, forecasting estimates may serve as early warning signals, triggering actions to bring the project back on track. Even though EVM allows for time and cost monitoring, initial research efforts were mainly directed to cost forecasting. An overview of the different forecasting methods and their accuracy can be found in Christensen (1993). In the early 2000s, the dominance of the cost objective persisted (see e.g. Fleming and Koppelman (2003) who discuss a project's price tag) until the introduction of the Earned Schedule concept by Lipke (2003). From this point onwards, the time dimension received growing attention, which culminated in publications on time forecasting (see Vandevoorde and Vanhoucke (2006)).

Dynamic scheduling aims at the integration of its three components. The first attempts at integrating schedule risk analysis and project control were executed by Vanhoucke (2010b) and Vanhoucke (2011). These research studies compare bottom-up (as found in Schedule Risk Analysis) and top-down (as found in EVM) project tracking approaches and study their relation to a project network's topological structure. Furthermore, activity sensitivity was incorporated in a dynamic corrective action framework. In a recent publication, Elshaer (2013) proposed an adaptation of one of the Earned Schedule forecasting methods using activity sensitivity metrics. By bridging top-down and bottom-up metrics, he was able to improve the forecasting accuracy of the Earned Schedule method. These publications formed the primary motivation for this paper's research. In order to construct sensitivity measures on the activity level, assumptions need to be made about the range and distribution of the activity durations. Using Monte Carlo simulations, various sensitivity measures can be constructed that provide an idea about the contribution of an activity to the project's overall sensitivity. However, each simulation run also yields top-down data that can be captured using the EVM performance metrics. This wealth of historical top-down data has great potential value in assisting project managers to make more accurate predictions and will be used by our proposed method. The contribution of this paper is threefold. First of all, we provide a clear framework of how a project manager can use the information from Monte Carlo simulations to improve project forecasting. The field of Artificial Intelligence, a research branch devoted to learning relations between attributes to construct one or multiple outputs, is ideally suited for this purpose. In this paper, we will focus on Support Vector Machines, a well-known technique for classification and prediction. Secondly, this paper intends to improve forecasting estimates using a computational experiment on a large and topologically rich dataset. In order to achieve this purpose, the forecasting accuracy is compared based on a large amount of runs and based on different scenarios. These scenarios provide valuable insights about when the proposed Support Vector Machine approach yields the biggest advantage. Finally, robustness checks are performed to illustrate the pitfalls of using historical data. This is particularly interesting since Artificial Intelligence is susceptible to the well-known "garbage-in, garbage-out" principle.

The outline of this paper is as follows. Section 2 provides a short overview of the underlying principles of Support Vector Regression. In section 3, the research methodology is outlined. The methodology consists of six steps, namely network generation, Monte Carlo simulation, attributes, the division between training and test set, cross-validation and grid search and finally, the testing phase. The settings of the computational experiment are delineated in section 4 using the six methodological steps. Section 5 presents the main results from the computational experiment and is broken down as follows. First, section 5.1 provides a thorough discussion of the finetuning process of the parameters of the Support Vector Regression. A link between the simulation scenario, topological structure and forecast accuracy is established. Next, the relation between accuracy and the project's point of completion is scrutinized. Finally, the limitations of our findings are discussed in section 5.3 which deals with a robustness check of the computational study. Section 6 draws conclusions and highlights future research avenues.

2. Support Vector Machine Regression

2.1. General theory

Support Vector Machines (SVM) in their current form were developed at the AT&T Bell Laboratories and gained momentum with the paper by Cortes and Vapnik (1995). Initial applications focused on binary classification of test instances and pattern recognition. With the rapidly increasing attention for SVMs, a number of introductory articles surfaced and constitute the foundation for this section (Burges (1998), Smola and Schölkopf (2004) and Mangasarian (2003)). In general, SVMs employ a model to construct a decision surface by mapping the input vectors into a high-dimensional (or infinite-dimensional) feature space. Next, a linear regression is executed in the high-dimensional feature space. This mapping operation is necessary because most of the time, the relation between a multidimensional input vector x and the output y is unknown and very likely to be non-linear. Support Vector Machine Regression (SVR) aims at finding a linear hyperplane, which fits the multidimensional input vectors to output values. The outcome is then used to predict future output values that are contained in a test set. Let us define a set of data points $P = (x_i, a_i)$,

$i = 1, ...n$ with x_i the input vector of data point i, a_i the actual value and n the number of data points. For linear functions f, the hyperplane that is constructed by the SVR is determined as follows:

$$f(x) = w\ x + b \tag{1}$$

Notation-wise, equation (1) displays similarities to a linear regression model. The predicted value, $f(x)$, depends on a slope w and an intercept b. In general, one wants to strike a balance between learning the relation between inputs and outputs while maintaining a good generalization behaviour. An excessive focus on minimizing training errors may lead to overfitting. A model with low complexity is limited with regard to the decision boundary it can produce but is less likely to overfit. In Cortes and Vapnik (1995), it is shown that the probability of a test error depends on two factors, namely the frequency of the training error and a confidence interval, where both factors form a trade-off. The confidence interval is related to the Vapnik-Chervonenkis dimension of the Support Vector Machine, which can be thought of as the complexity of the learning model. Hence, improved generalization may be obtained by improving the confidence interval at the expense of additional training errors. The primary instrument to control this trade-off is C, which explains its importance. The balance between good training and generalization behaviour is reflected in equation (2), where R denotes the compound risk caused by training errors and model complexity. Naturally, the risk R needs to be kept as low as possible.

$$R = \frac{C}{n}\sum_{i=1}^{n} L_\epsilon(a_i, f(x_i)) + \frac{1}{2}||w||^2 \tag{2}$$

Equation (2) yields estimated values for w and b and consists of two main parts. The first part, $\frac{C}{n}\sum_{i=1}^{n} L_\epsilon(a_i, f(x_i))$ consists of the training or empirical risk and is measured by the ϵ-insensitive loss function, $L_\epsilon(a, y)$ (see e.g. Vapnik (1998)). This function implies that the prediction error is ignored if the difference between the predicted value $f(x)$ and the actual value a is smaller than ϵ. The ϵ-insensitive loss function is formally defined in equation (3).

$$L_\epsilon(a, y) = \begin{cases} |a - f(x)| - \epsilon\ |a - f(x)| & \geq \epsilon \\ 0 & otherwise \end{cases} \tag{3}$$

The second part of equation (2), $\frac{1}{2}||w||^2$, is the regularization term and is related to the complexity of the model (see Cortes and Vapnik (1995)). C controls the trade-off between the regularization term and the training accuracy. Large values of C imply that more weight is put on correctly predicting training points, at the cost of a higher generalization error.

The problem of finding an optimal hyperplane is a convex optimization problem. For non-linear relations between input vectors and outputs, it is necessary to define a map, ϕ, that translates the training points x_i into a higher-dimensional feature space. The consequence is that w, after constructing a Lagrangean function from (1) will no longer be a function of x_i but of $\phi(x_i)$ and that

Kernel name	Formula
Linear	$x_i^T x$
Polynomial	$(\gamma x_i^T x + r)^d$
Radial Basis	$e^{-\gamma(\|\|x_i - x\|\|^2)}$
Sigmoidal	$tanh(\gamma x_i^T x + r)$

Table 1: Overview of common kernel functions

the product $\phi(x_i)\phi(x)$ needs to be calculated. We refer the reader to Smola and Schölkopf (2004) for full details about these observations. The function $\phi(x_i)\phi(x)$ is often defined as $K(x_i, x)$ and is referred to as a kernel function. Kernel functions try to achieve linear separability between training points in the higher-dimensional feature space. Many kernel functions exist. In fact, any function that satisfies Mercer's condition (Vapnik (2000)) can serve as a kernel function. An overview of frequently occurring kernel functions is given in table 1. γ, r and d are parameters that are kernel-specific. It is worth noting that the Radial Basis Function kernel (RBF) is sometimes parameterized using $\frac{1}{\delta^2}$ instead of using γ.

2.2. Application to project control

To the best of our knowledge, only two papers deal with SVMs applied to Earned Value Management. Both of them are inspired from a practical point of view, rather than pertaining to a more general academic context. The first paper, by Cheng et al. (2010), employs a SVM model that is tuned by means of a fast messy genetic algorithm with the goal of estimating the final cost of two construction projects. The combination of the genetic algorithm and the SVM was fused with fuzzy logic in a second paper by Cheng and Roy (2010). The proposed hybrid system was tested on an artificial problem (function approximation) and two real-life problems, namely a conceptual cost estimation and the estimate at completion.

3. Methodology

The Support Vector Regression, explained in the previous section, will be applied to a project control environment in order to construct more reliable time and costs forecasts, using periodic EVM data as inputs to learn from. In this section, an overview of the methodology that was employed will be given. It is comprised of six distinct steps, namely network generation, Monte Carlo simulation, attributes, the division into a training and test set, cross-validation and grid search and finally, the testing phase. The goal of this section consists of showing the structured process by which the Support Vector Regression gains knowledge about the attributes in order to construct a more accurate forecast in the testing phase. We will now discuss each part of the methodology and refer to figure 1 for a schematic overview.

Network Generation. In the first phase, a large number of project networks are generated with the aim of constructing a dataset that is topologically diverse. The topological structure of the networks is based on the SP-factor[1], which provides a degree of closeness to a completely serial or parallel network. Low values of the SP-factor are associated with parallel networks, while the opposite observation holds for serial networks. Furthermore, the construction of a dataset requires generating activity durations and costs. The project is scheduled in order to construct a baseline schedule that will be used as a reference point for the project control data. Information of the progress of the project will be compared against the baseline schedule in order to determine deviations in terms of time and cost. These deviations will be generated using Monte Carlo simulations, which is the second step of the methodology.

Monte Carlo simulation. The second step of the methodology uses Monte Carlo simulations. These enable us to introduce time and cost variability on the activity level. The process of Monte Carlo simulation proceeds as follows. First of all, a probability distribution is constructed on which the activity duration uncertainty is based. In this paper, we use the generalized beta distribution. Not only is this distribution used in previous research studies (e.g. Vanhoucke (2010b)), it is also applied to real-life situations, including construction project simulations (AbouRizk et al. (1994)). The generalized beta distribution is a continuous probability distribution parameterized using a lower limit a, an upper limit b and two shape parameters, θ_1 and θ_2. The probability density function can be expressed as follows, where $\Gamma(\cdot)$ refers to the gamma function:

$$f(x) = \frac{\Gamma(\theta_1 + \theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)(b-a)^{\theta_1+\theta_2-1}}(x-a)^{\theta_1-1}(b-x)^{\theta_2-1}, x \in [a,b] \tag{4}$$

Historically, the triangular distribution is often preferred to the beta distribution because of its straightforward nature or as an approximation for the beta distribution (Johnson (1997)). However, Kuhl et al. (2007) warn against the use of the triangular distribution in the absence of empirical datapoints. The authors argue that the beta distribution is preferred to the triangular distribution when the distribution of the random variable is clearly skewed to the left or to the right. In our experiment, we assume that values for the lower limit a, upper limit b and mode c are given. Kuhl et al. (2007) have shown that by using an auxiliary quantity, the beta density function approximates the mode c very accurately. This is achieved by using the quantity $r = \frac{b-c}{c-a}$ to estimate the shape parameters of the generalized beta distribution. The shape parameters are then given by:

$$\theta_1 = \frac{r^2+3r+4}{r^2+1} \quad \theta_2 = \frac{4r^2+3r+1}{r^2+1} \tag{5}$$

[1] Even though the SP-factor is originally named the I_2 indicator, it is commonly referred to as the SP-factor (see e.g. Vanhoucke et al. (2008).

Using a, b, θ_1 and θ_2, a wide array of shapes for the generalized beta distribution can be generated. Consequently, this distribution is ideally suited to construct scenarios where projects finish earlier, later or on time compared to the baseline schedule.

Attributes. The Monte Carlo simulations are used to generate deviations from the baseline schedule, resulting in measures that are able to capture the degree to which deviations occur. Input measures for machine learning algorithms are often referred to as attributes. In a project control environment, the attributes correspond with EVM measures and play a crucial role in the context of this paper. The SVR model learns the relation between the attributes and the output values. It is worth pointing our that the attributes correspond with the multi-dimensional input vector x_i of section 2.1, where i indexes the training instance. The output value, a_i, refers to the Real Duration for time forecasting and to the Real Cost for cost forecasting. The SVR model tries to predict a_i using the attributes, x_i that are given in table 2. The attributes shown at the top of the table, SPI, SPI(t), CPI and ES are captured for time and cost forecasting. Naturally, the time forecasting methods are specific for project duration forecasting. They are denoted as the Estimated time At Completion (EAC(t)), in which a subdivision is made according to the Planned Value (PV), Earned Duration (ED) and Earned Schedule (ES) method. The same reasoning applies to the cost forecasting methods, which are denoted as the Estimated cost At Completion (EAC). AD and PD denote the Actual Duration and Planned Duration, whereas the Budget At Completion is abbreviated to BAC. The final time forecasting method, $EAC(t)_{ES_2\alpha}$, finds its roots in a paper by Elshaer (2013), who proposed an adaptation of the PV and EV based on sensitivity information:

$$PV'_{\alpha,t} = \sum_j \alpha_j PV_{j,t} \tag{6}$$

$$EV'_{\alpha} = \sum_j \alpha_i EV_{j,AT} \tag{7}$$

In these equations, i indexes the activities and α_i refers to the value of sensitivity metric α for activity i. In order to preserve consistency, the sensitivity metrics found in Vanhoucke (2010a) and Elshaer (2013) were adopted. Six different sensitivity metrics are employed, namely the Criticality Index (CI), the Significance Index (SI), the Schedule Sensitivity Index (SSI) and the Cruciality Index (CRI) using Pearson's product moment (CRI$_r$), Spearman's rank correlation (CRI$_\rho$) and Kendall's τ rank correlation (CRI$_\tau$). Because of this change in Planned Value and Earned Value, the numbers for the Earned Schedule and Schedule Performance Indicator change as well. These changes are reflected in table 2, where ES$'$ and SPI(t)$'$ point out this difference and α denotes the sensitivity metric ($\alpha \in$ {CI, SI, SSI, CRI$_r$, CRI$_\rho$, CRI$_\tau$}). For further details with regard to the formulas of table 2, the reader is referred to Vanhoucke (2014) and Elshaer (2013). As the project progresses, different values for the performance indicators can be observed. Hence, every attribute will be captured across different review periods. T refers to the total number of reporting periods and is indexed by rp ($rp = 1, 2, ...T$).

Attribute		Calculation	
SPI		$\frac{EV}{PV}$	
SPI(t)		$\frac{ES}{AT}$	
CPI		$\frac{EV}{AC}$	
ES		$t + \frac{EV - PV_t}{PV_{t+1} - PV_t}$	
Time		Cost	
EAC(t)$_{PV_1}$	$PD - \frac{(EV-PV)*PD}{BAC}$	EAC$_1$	$AC + (BAC - EV)$
EAC(t)$_{PV_2}$	$\frac{PD}{SPI}$	EAC$_2$	$AC + \frac{BAC-EV}{CPI}$
EAC(t)$_{PV_3}$	$\frac{PD}{CPI*SPI}$	EAC$_3$	$AC + \frac{BAC-EV}{SPI}$
EAC(t)$_{ED_1}$	$PD + AD * (1 - SPI)$	EAC$_4$	$AC + \frac{BAC-EV}{SPI(t)}$
EAC(t)$_{ED_2}$	$\frac{PD}{SPI}$	EAC$_5$	$AC + \frac{BAC-EV}{SCI}$
EAC(t)$_{ED_3}$	$\frac{PD}{SPI*CPI} + AD * (1 - \frac{1}{CPI})$	EAC$_6$	$AC + \frac{BAC-EV}{SCI(t)}$
EAC(t)$_{ES_1}$	$AD + PD - ES$	EAC$_7$	$AC + \frac{BAC-EV}{0.8CPI+0.2SPI}$
EAC(t)$_{ES_2}$	$AD + \frac{PD-ES}{SPI(t)}$	EAC$_8$	$AC + \frac{BAC-EV}{0.8CPI+0.2SPI(t)}$
EAC(t)$_{ES_3}$	$\frac{PD-ES}{CPI*SPI(t)}$		
EAC(t)$_{ES_2\alpha}$	$AD + \frac{PD-ES'}{SPI(t)'}$		

Table 2: Overview of the attributes of the SVM

Training and Test set. The previous steps of the methodology ensure that a large and diverse dataset of projects is constructed. This set is then decomposed into a training and test set. Normally, this process is structured as follows. A distinction between a training, validation and test set is made. The training set is used to allow the SVR model to learn the relation between x_i and a_i based on the Monte Carlo simulations defined in the second step of the methodology. The validation set consists of examples that are mainly used to tune the parameters of the model. The model with the tuned parameters is then applied to the test set to assess the performance. Within our setting, this would imply that for every project, the data that results from the Monte Carlo simulations are divided into a training, validation and test set. Hence, model parameters would be tuned on a project level. Since this paper aims to provide recommendations that are as general as possible, tuning parameters on the project level limits the applicability of the findings of this paper. Hence, a deviation is proposed that functions on a more aggregated level. The dataset is partitioned in a training set, consisting of 10% of the total number of projects, whereas the test set contains the remaining 90%. Immediately, it can be seen that the training and test set are composed across projects rather than within projects. The training set is then further divided into two sets, a smaller training set and a validation set, using cross-validation.

Cross-validation and Grid Search. A vital issue of an Artificial Intelligence method concerns the tuning of parameters. For a Support Vector Machine, the parameters depend on the kernel choice.

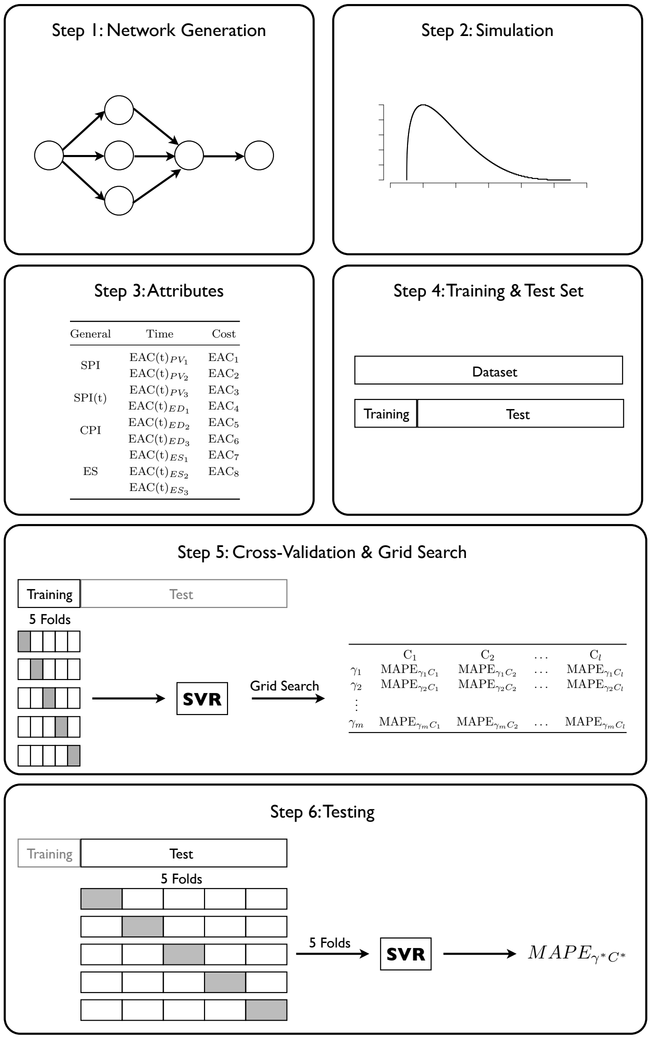
In section 2.1, the kernel functions and their most prominent identities were discussed. In this paper, we opted for the Radial Basis Function kernel. This choice was inspired by three reasons (Hsu et al. (2003)). First of all, the RBF kernel can map a non-linear relationship between attributes and outputs, which is not the case for the linear kernel. Additionally, Keerthi and Lin (2003) have shown that the linear kernel is a special case of the RBF kernel. Secondly, the RBF kernel counts fewer parameters than the polynomial kernel. Finally, there are little numerical difficulties involved in the use of the RBF kernel. Every kernel has specific (hyper)parameters that need to be estimated. Based on a training set, the two RBF parameters, C and γ, need to be set in such a manner that the SVR model can correctly predict the project duration of new data instances, which comprise the testing set. In general, there is little guidance as to determining the parameter values, except for the approach of Hsu et al. (2003), who advocate the use of a grid search procedure combined with cross-validation.

In our study, cross-validation according to the "leave-one-out" principle was applied. k-fold cross-validation implies partitioning the training set into k folds of equal size, where each of the k folds is used as a validation set and the other $k-1$ folds serve as training instances. Thus, each instance is predicted once. The prediction results are averaged across the different folds to assess the performance. The rationale behind cross-validation consists of eliminating the probability of overfitting and strengthening the generalization ability of the regression. Employing a grid search indicates that different combinations of C and γ are tried and that the combination with the best score across the k folds is elected. In figure 1, a subdivision of the training set is made using 5 folds. Each time, the training set is fed to the SVR model. After it has learned the link between the attributes and real duration or real cost, predictions are made using the validation set. For every combination of C and γ, the predictive power can be assessed. Suppose there are l different levels for C and m different levels for γ. It is possible to list the accuracy and select the parameters that yield the highest accuracy. Forecasting accuracy is measured using the Mean Absolute Percentage Error (MAPE), which has been used in previous forecasting studies (Vanhoucke and Vandevoorde (2007)) and is given by:

$$MAPE = \frac{1}{T} \sum_{rp=1}^{T} \frac{|a_i - EAC(t)_{rp}|}{a_i} \tag{8}$$

for time forecasting. For cost forecasting, $EAC(t)_{rp}$ is replaced by EAC_{rp}, the cost estimate at completion. Equation (8) shows that the percentage deviation from the actual value a_i is averaged across all review periods. The MAPE can be used as an indicator to find the optimal values of C and γ throughout the grid search. The best found values are denoted by C^* and γ^*, respectively.

Testing. Finally, the testing phase concludes the methodology. Using C^* and γ^*, the SVR model is applied to the remaining 90% of the entire dataset. In order to mitigate possible bias of drawing 1 sample for training and 1 for testing, 5 folds are used. Consequently, 80% of the Monte Carlo simulations containing periodic EVM data of an individual project is used for training, after which the SVR with parameter values C^* and γ^* predicts the real duration and real cost of the remainder of the test set. The MAPE is then averaged across the 5 folds.



General	Time	Cost
SPI	EAC(t)$_{PV_1}$ EAC(t)$_{PV_2}$	EAC$_1$ EAC$_2$
SPI(t)	EAC(t)$_{PV_3}$ EAC(t)$_{ED_1}$	EAC$_3$ EAC$_4$
CPI	EAC(t)$_{ED_2}$ EAC(t)$_{ED_3}$	EAC$_5$ EAC$_6$
ES	EAC(t)$_{ES_1}$ EAC(t)$_{ES_2}$ EAC(t)$_{ES_3}$	EAC$_7$ EAC$_8$

	C$_1$	C$_2$	...	C$_l$
γ_1	MAPE$_{\gamma_1 C_1}$	MAPE$_{\gamma_1 C_2}$	...	MAPE$_{\gamma_1 C_l}$
γ_2	MAPE$_{\gamma_2 C_1}$	MAPE$_{\gamma_2 C_2}$	...	MAPE$_{\gamma_2 C_l}$
$\vdots$				
γ_m	MAPE$_{\gamma_m C_1}$	MAPE$_{\gamma_m C_2}$	...	MAPE$_{\gamma_m C_l}$

Figure 1: Overview of the 6 phases of the methodology

4. Computational Experiment

In this section, an explanation of the computational experiment is given. The six steps that were outlined in the methodology will now be specified by providing the settings that were utilized. The steps are listed in the same order as was done in the previous section.

Network Generation. For this study, 900 Activity-on-the-Node (AoN) networks were used with 30 activities and random activity durations and costs. As mentioned in section 3, the topological structure of the networks is based on the SP-factor. The AoN networks of this computational study vary from 0.1 to 0.9 in steps of 0.1. Consequently, for every level of the SP-factor, 100 networks were generated. This was done using the RanGen engine (Demeulemeester et al. (2003) and Vanhoucke et al. (2008)). The dataset of this study has been used in previous computational studies (see e.g. Vandevoorde and Vanhoucke (2006) and Vanhoucke and Vandevoorde (2007)) and is available on `www.projectmanagement.ugent.be/evms.html`. Apart from the topological structure of the network, baseline costs and durations for every activity need to be determined. These were generated randomly between 50 and 100 (for the costs) and 20 and 40 (for the durations), respectively. It is worth remarking that the costs are entirely variable, implying that if a project deviates from its plan, the cost deviation varies completely in line with the deviation in duration. The underlying assumption is that the activity costs are expressed in monetary units per time unit. Hence, if an activity takes longer to finish, it will require more man-hours and the associated costs will rise.

Monte Carlo simulation. The Monte Carlo simulations enable us to introduce time and cost variability on the activity level, which will be reflected in the EVM metrics at the project level. The process of Monte-Carlo simulation proceeds as follows. The three input parameters (a, b and c) that were selected resulted into 6 different execution scenarios, which are summarized in table 3 and are described along the following lines. Generally, 3 situations can be discerned where early (real duration (RD) $<$ planned duration (PD)), late (RD $>$ PD) or on time situations (RD $\approx$ PD) arise. For each of these three situations, two instances are generated. The scenarios with a lower variability are indexed using subscript 1 in table 3, whereas the subscript 2 is used for scenarios where the variability is larger. The last three columns provide the lower limit a, mode c and upper limit b as a percentage of the baseline duration of activity j, denoted by d_j. As mentioned in section 3, a, b and c give rise to the auxiliary quantity, which in turn enables us to determine the shape parameters θ_1 and θ_2.

Attributes. The attributes, which were described in section 3, are captured at different points in time. In order to make a comparison across projects possible, the values were captured for every 10% complete of the project, ranging from 10% to 90%. Therefore, T = 9 in equation (8).

Training and Test set. During the network generation phase, 100 projects per level of the SP-factor are generated. 10 projects (10%) are used for the training phase, whereas 90 projects (90%) are used for testing purposes. In order to compute the sensitivity metrics, 1,000 runs of every project are executed. The sensitivity metrics are then used to calculate $\text{PV}'_{\alpha,t}$ and EV'_{α} according to equations (6) and (7). Hence, every project is re-executed 1,000 times using the generalized beta distribution, after which the different folds for cross-validation can be constructed. As was mentioned in section 3, a different approach with regard to the training and test set is used. Let us name the traditional

Scenario	Abbreviation	Settings		
		a	c	b
Early	E$_1$	0.5 d_j	d_j	1.1 d_j
	E$_2$	0.1 d_j	d_j	1.1 d_j
On Time	OT$_1$	0.8 d_j	d_j	1.2 d_j
	OT$_2$	0.5 d_j	d_j	1.5 d_j
Late	L$_1$	0.9 d_j	d_j	1.5 d_j
	L$_2$	0.9 d_j	d_j	1.9 d_j

Table 3: Overview of the settings of the 6 different scenarios

process of training and test the project approach. In the project approach, the 1,000 executions of the project under study would be divided into 80% of the data (800 executions) for training and the remainder (200 executions) for testing. The training set would be partitioned into a smaller training set, containing 80% of the executions (0.8 * 800 = 640 executions) and 20% for validation (0.2 * 800 = 160 executions). The goal of these latter two sets is to determine the best parameters using a grid search. Using the best found parameters, C^* and γ^*, the model would be retrained on the entire training set (800 executions) and tested on the test set (200 executions). However, this implies that the parameters are set on a project basis, which is at odds with our goal of making generally applicable recommendations. As a result, the process of training, validation and testing was adapted. That is why 90 projects (10 projects per SP-factor level) were used for determining the best parameters. For each of those 10 projects, 800 executions were used for training and 200 for validation purposes. In this phase, a grid search for the best parameter values is conducted. Using cross-validation, this process is repeated a number of times and the parameter settings that yield the best results across all folds and all projects will be used for the 810 projects (90 projects per SP-factor level) of the test set. For each of those 90 projects, 800 executions are used to learn the relationship between the attributes of table 2 and the real duration or real cost. This set was named the learning set to indicate that it differs from the training set and to avoid confusion. The remaining 200 executions represent the actual runs of the project. Hence, it can be seen that the main difference lies in the training phase where we opted to find parameter values on a higher level than that of individual projects. The distinction between the training and test set is summarized in table 4.

Cross-validation and Grid Search. In this paper, k-fold cross-validation was employed. 5 folds were used, which implies that each time 800 runs are used for training and 200 runs are used for

Data Instances		
projectmanagement.ugent.be/evms.html	SP-indicator	0.1-0.9, Δ_{SP}=0.1
900 (9 * 100) project instances	#projects	100 per SP level
Training Set		
Find optimal C and γ		
90 (9 * 10) project instances	Training Set	800 executions
Project 1-10 per SP level	Validation Set	200 executions
1,000 executions/project	Repeated 5 times	5 folds
Test Set		
Test performance on unseen data using C^* and γ^*		
810 (9 * 90) project instances	Learning Set	800 executions
Project 11-100 per SP level	Test Set	200 executions
1,000 executions/project	Repeated 5 times	5 folds

Table 4: Overview of the training and test set

validation. It can be seen that cross-validation counters overfitting since every run is used exactly once for validation. The training data is fed to the SVR model, which then constructs a model that is applied to the validation set. However, the main goal of validation consists of finding the optimal parameter settings for the RBF kernel. For the computational experiment, exponentially growing sequences of C ($= 2^{-5}, 2^{-3}, ..., 2^{13}, 2^{15}$) and γ ($= 2^{-15}, 2^{-13}, ..., 2^{1}, 2^{3}$) are tried. The combination of C and γ that yields the lowest MAPE will then be used for the test set.

Testing. The outcome of the previous step is an optimal value for C (denoted by C^*) and γ (denoted by γ^*). Using these parameter values, the test set which consists of 90 projects is divided into learning and test data, as shown in table 4. Again, 80% (800 runs) is used for learning the relation between the attributes and output and 20% (200 runs) is used for testing. In order not to introduce any bias, 5 folds are constructed again. The advantage of implementing these folds in the testing phase is that every Monte Carlo simulation is used exactly once for testing. The results are averaged across the 5 folds and culminate into a final MAPE, denoted by MAPE$_{\gamma^* C^*}$ in figure 1.

5. Results

In this section, a comparison is made between the SVR model and the time and cost forecasting methods that are part of the list of attributes listed in table 2. The support vector regression was implemented using the LIBSVM library in R (Chang and Lin (2011)) and tested on Ghent

University's High Performance Computing infrastructure. More specifically, the computational experiment was run on the Delcatty cluster, which has 64 GB RAM available and makes use of a quad-core Intel Xeon processor with 2.6 GHz.

In the remainder of this section, the MAPE will be reported as a criterion for forecasting accuracy. Hence, if the MAPE is reported in relation to the period, this reflects the Absolute Percentage Error, averaged across all periods up until the period under study. This must be interpreted with care since the MAPE implies that a method at a certain period is better *on average*, rather than claiming that a method reaches a better value in that specific time instance, as would be the case for the Absolute Percentage Error.

The structure of this section is as follows. Section 5.1 reveals the optimal parameter settings C^* and γ^* for the Support Vector Regression model. In section 5.2, the performance of SVR is compared against 15 EVM time forecasting methods and 8 cost forecasting methods. First of all, the forecasting accuracy is discussed in light of the SP-factor, after which the results are refined along the execution scenario. Section 5.2 also examines the performance in function of the percentage complete, which was varied from 10% to 90% (cf. section 4). Finally, section 5.3 concludes with some robustness checks in order to illustrate the limitations of the proposed Artificial Intelligence method.

5.1. Parameter fine-tuning

For the outcome of the 5-fold cross-validation, it is possible to make a distinction between the 9 different SP-factors, the 6 execution scenarios and whether time or cost predictions are made. All of these settings were detailed in section 4. However, in many situations, a project manager has incomplete information with regard to the distribution parameters of the project's progress. The only element that can be determined with certainty is the SP-factor. The SP-factor is a static indicator, related to the topology of a project's network and can be assessed prior to the project's execution. As a result, identifying the best parameter settings based on incomplete information is a realistic and practical requisite before making a comparison with the current best forecasting methods. We found that, regardless of the SP-factor and scenario, the best value for γ equals 2^{-15}. Setting the second hyperparameter of the RBF kernel requires more nuancing, since the optimal values range from 2^5 to 2^{11} for time forecasting and from 2^7 to 2^{13} for cost forecasting. The main question in finding a good value for C is whether all parameter settings are equally important. Are the ramifications of setting a non-optimal C the same for different values of SP and depending on the scenario?

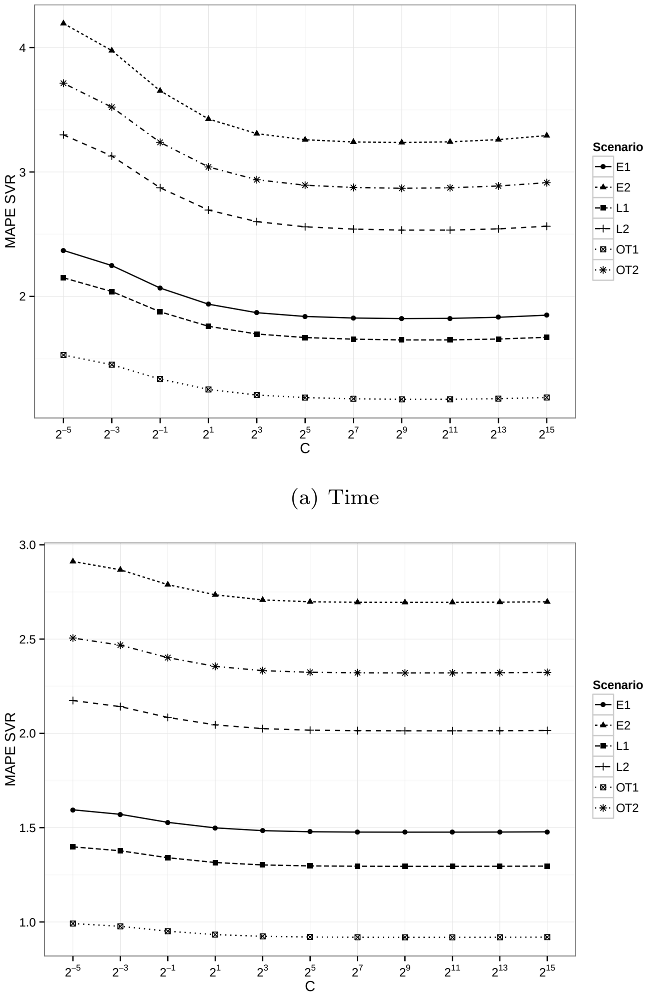
In order to answer this question, we turn towards figures 2(a) and 2(b). In these two figures, the cost parameter is displayed on the x-axis and the MAPE of the SVR is given on the y-axis. Since there was little difference between the different SP-factor levels, the MAPE was averaged across this topological indicator. It is clear that the MAPE stabilizes from a value of 2^5 onwards.

	Parameter	Frequency
Time	2^5	11.11%
	2^7	22.22%
	$\mathbf{2^9}$	**46.3%**
	2^{11}	20.37%
Cost	2^7	24.07%
	$\mathbf{2^9}$	**50%**
	2^{11}	9.26%
	2^{13}	16.67%

Table 5: Frequency of the optimal parameter value for C across the 6 scenarios and 9 SP-factor levels

Consequently, the cost parameter with a maximum frequency in table 5 is selected. This table lists the optimal parameter value for C for time and cost forecasting, along with their frequency. For time and cost forecasting, C is set to 2^9. The maximum MAPE error that is made by choosing this non-optimal parameter value is on average equal to 0.008% and 0.003% for time and cost forecasting, respectively.

Concluding, the best found parameter settings are $C = 2^9$ and $\gamma = 2^{15}$ for predicting a project's final duration and cost. In the following subsection, the forecasting accuracy for the SVR model with these parameter settings will be compared with the 15 time forecasting methods and the 8 cost forecasting methods.



(a) Time

(b) Cost

Figure 2: Effect of C on the time 2(a) and cost 2(b) performance

5.2. General Performance

In the following paragraphs, the effect of the SP-factor, the execution scenario and the percentage complete is studied.

Effect of the SP-factor. The SP-factor is known to have a significant impact on the results of the 9 traditional forecasting methods, with a reported increase in forecasting accuracy as the SP-factor rises (Vanhoucke (2011)). Compared to the Critical Path Method, EVM reports results on a more

Criterion	PV	ED	ES	Elshaer	SVR
PF	SPI	SPI	SPI(t)	SI	
μ_{MAPE}	4.32%	4.31%	3.63%	3.72%	2.21%
σ_{MAPE}	1.86%	1.85%	1.60%	1.58%	1.11%
$\Delta_{\mu_{MAPE}}$	2.79%	2.82%	3.17%	2.97%	2.39%

Table 6: Overview of the time forecasting results based on the SP-factor

Criterion	EAC_1	EAC_2	EAC_3	EAC_4	EAC_5	EAC_6	EAC_7	EAC_8	SVR
μ_{MAPE}	3.93%	2.16%	2.68%	2.52%	5.11%	5.19%	2.17%	2.17%	1.28%
σ_{MAPE}	2.63%	0.82%	1.10%	0.96%	2.34%	2.29%	0.82%	0.82%	2.03%
$\Delta_{\mu_{MAPE}}$	0.09%	0.57%	1.16%	0.93%	0.87%	0.62%	0.59%	0.58%	0.40%

Table 7: Overview of the cost forecasting results based on the SP-factor

aggregated level of the Work Breakdown Structure. As a result, it is possible that critical activities are delayed while EVM metrics report that everything is fine because non-critical activities are ahead of schedule. This criticism has also been formulated by Jacob and Kane (2004), yet we maintain that executing control on a higher WBS level is the only practical solution for project managers, rather than keeping track of every individual activity's progress. However, for more serial projects, almost every activity is critical and the findings of CPM and EVM converge, which means that EVM will be less susceptible to report false warning signals. Upon examining the relation between the MAPE and the SP-factor, we were able to corroborate these findings. Tables 6 and 7 summarize the results of the time and cost forecasting methods, respectively. The row labeled "PF" displays the Performance Factor of the best performing forecasting method. For the methods proposed by Elshaer (2013), the performance factor corresponds with the sensitivity measure on which the forecast is based. μ_{MAPE} is the mean MAPE across all levels of the SP-factor, while σ_{MAPE} provides results for the standard deviation. Finally, $\Delta_{\mu_{MAPE}}$ is calculated as the maximum difference in MAPE between SP-levels. A number of conclusions can be drawn from these tables:

- The general performance of Support Vector Regression is very encouraging. In tables 6 and 7 it is shown that our newly proposed method outperforms the current EVM methods. In table 6, the average improvement amounts to 1.42% compared to the incumbent method, ES_2. For the cost dimension, the difference in performance compared to EAC_2 is equal to 0.88%. It is worth noting that the MAPE percentages (and the difference in performance) are lower for predicting the final cost compared to project duration forecasting.

- The SVR method reports the lowest standard deviation across all time forecasting methods ($\sigma_{MAPE} = 1.11\%$). For cost forecasting, the variability of the SVR method is larger. EAC$_2$, EAC$_7$ and EAC$_8$ have the smallest standard deviation. It is worth pointing out that this slightly smaller standard deviation needs to be contrasted with the worse performance for μ_{MAPE}.
- The row with $\Delta_{\mu_{MAPE}}$ as its criterion can be calculated as the maximum difference in mean MAPE across SP levels. For the time forecasting methods, SVR shows the characteristic improvement in performance as the project becomes more serial, which means that $\Delta_{\mu_{MAPE}}$ could be calculated as $\mu_{MAPE,SP=0.1} - \mu_{MAPE,SP=0.9}$. The forecasting improvement in relation to the SP-factor is shown in figure 3 and corroborates the research of Vanhoucke (2010a). This figure displays the SP-factor on the x-axis and the MAPE on the y-axis. As expected, the methods proposed by Elshaer (2013) attain better results for parallel projects than the traditional ES method and remain competitive for the other SP-levels. The relation between the SP factor and MAPE does not hold for cost forecasting.

The results of table 6 and 7 were also checked statistically. Using the non-parametric Mann-Whitney U Test, it is possible to check whether the location of the SVR method and the incumbent time (ES$_2$) and cost (EAC$_2$) method displays a shift in location. The alternative hypothesis states that the location shift is not equal to 0. The results of the Mann-Whitney U Test reveal that the null hypothesis can be rejected ($p < 2.2e^{-16}$) at the 95% confidence level, indicating that there is a location shift.

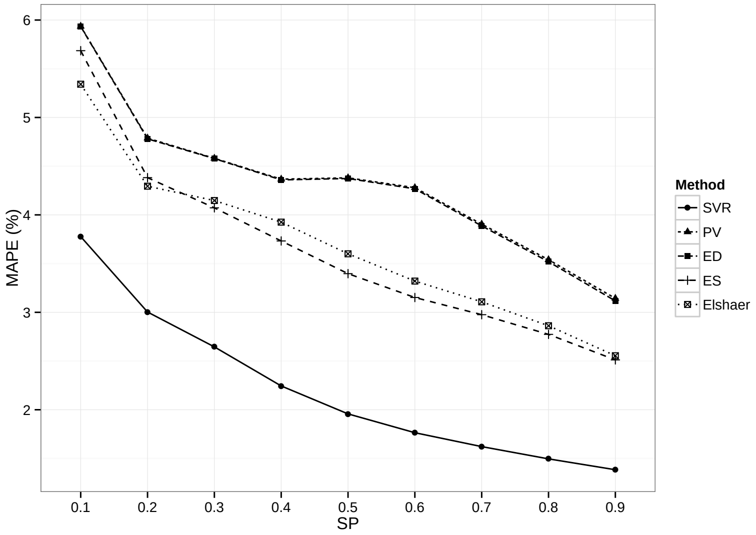


Figure 3: Relation between the SP-factor and MAPE for the time forecasting methods

Scenario	Criterion	PV	ED	ES	Elshaer	SVR
E_1	PF	SPI	SPI	SPI(t)	SI	
	μ_{MAPE}	3.57%	3.57%	2.99%	3.08%	1.83%
E_2	PF	SPI	SPI	SPI(t)	SI	
	μ_{MAPE}	6.74%	6.74%	5.40%	5.51%	3.22%
OT_1	PF	1	1	1	CRI_τ	
	μ_{MAPE}	1.39%	1.40%	1.33%	1.97%	1.17%
OT_2	PF	1	1	1	SI	
	μ_{MAPE}	3.34%	3.35%	3.20%	4.80%	2.86%
L_1	PF	SPI	SPI	SPI(t)	SI	
	μ_{MAPE}	3.33%	3.33%	2.67%	2.75%	1.64%
L_2	PF	SPI	SPI	SPI(t)	SI	
	μ_{MAPE}	5.57%	5.50%	4.11%	4.20%	2.53%

Table 8: Overview of the time forecasting results based on the scenario

Effect of the scenario. The forecasting accuracy discussed in the previous paragraph made no distinction between the scenarios that were used for the Monte Carlo simulations. In this paragraph, the MAPE is averaged across all SP-factor levels but insights into the relation between forecasting accuracy and execution scenario are given. The performance factor and average MAPE are reported in table 8 for time forecasting and in table 9 for cost forecasting. The following observations can be made:

- The Support Vector Regression model outperforms the time and cost forecasting methods across all scenarios. For the time dimension, the improvement compared to the second-best method ranges from 0.16% (OT_1) to 2.18% (E_2). For cost forecasting, the difference lies between 0.03% (OT_1) and 1.42% (E_2).

- The forecasting methods with a performance factor equal to 1 perform best for the OT scenarios. In the other scenarios, the time forecasting methods using the SPI or SPI(t) perform best, whereas the cost forecasting method with CPI as its performance factor outperforms the other traditional forecasting methods.

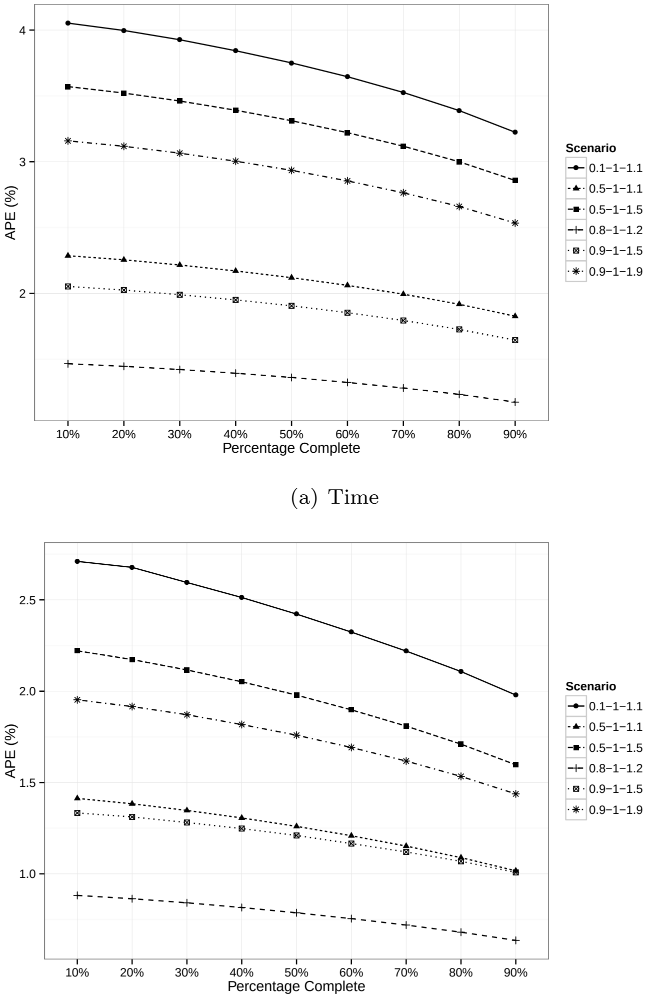
- All methods suffer from the higher variability that is present in the scenarios with the suffix 2. The highest MAPE is noted for scenarios E_2, OT_2 and L_2. This finding holds for both time and cost forecasting. It is worth mentioning that the SVR method is least susceptible to the increased variability. When computing the difference in μ_{MAPE} between scenarios with suffix 2 and those with suffix 1, the SVR method reports the smallest difference compared to the other methods.

Effect of the % Complete. As mentioned before, every 10% complete, from 10% to 90%, a measurement of the attributes was executed and a prediction was made. Hence, these progress predictions

Scenario	EAC_1	EAC_2	EAC_3	EAC_4	EAC_5	EAC_6	EAC_7	EAC_8	SVR
E_1	3.50%	1.78%	2.21%	2.07%	4.17%	4.21%	1.79%	1.78%	1.02%
E_2	8.32%	3.40%	4.17%	3.87%	8.38%	8.57%	3.41%	3.40%	1.98%
OT_1	0.67%	1.08%	1.29%	1.28%	1.99%	1.98%	1.09%	1.09%	0.64%
OT_2	1.67%	2.82%	3.26%	3.29%	5.05%	5.08%	2.83%	2.83%	1.60%
L_1	3.13%	1.51%	1.97%	1.80%	3.87%	3.89%	1.53%	1.53%	1.01%
L_2	6.28%	2.37%	3.17%	2.81%	7.18%	7.41%	2.40%	2.38%	1.44%

Table 9: Overview of the cost forecasting results based on the scenario

allow a decision maker to assess the forecasting performance as a function of the percentage complete. The results are depicted in figure 4(a) and 4(b). In these figures, the percentage complete is shown on the x-axis, while the Absolute Percentage Error (APE) is shown on the y-axis. Both figures reveal that the forecasting performance improves as more information about the project is known. This behaviour demonstrates similarities with the other forecasting methods, from which it was shown in Vanhoucke (2010a) that the forecasting performance improves as the percentage complete increases.



(a) Time

(b) Cost

Figure 4: Relation between the forecasting performance and the percentage complete for time (4(a)) and cost (4(b))

5.3. Robustness Checks

In the previous section, the influence of the SP-factor, scenario and percentage complete were examined under the assumption that the project manager is able to correctly identify the test set. We have demonstrated that the SVR outperforms the other techniques in these circumstances. In this paragraph, we relax this assumption and study the effects when the training set differs from the test set. The same parameters that resulted from section 5.1 are used for this experiment.

The robustness checks were run on a smaller set of the data, consisting of 10 projects with an SP-factor of 0.5. An overview of the similarities between the training and test set is given in figure 5. Section 5.2 revealed the superior performance of the SVR when the training and test set follow the same distribution with equal parameter settings. In this section, two additional situations will be examined. First of all, we study the change in performance when the training and test set distributions differ to a large degree. This is the subject of section 5.3.1. Secondly, section 5.3.2 presents additional scenarios using the 6 variants found in section 4. These represent situations in which the training set is similar but not equal to the test set.

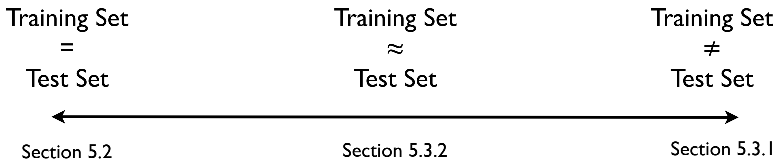


Figure 5: Overview of the similarities between the training and test set

5.3.1. Training set ≠ Test set

In this section, the training set is allowed to differ substantially from the test set. Each of the 6 scenarios is used as a training set and test set, leading to 36 (6 * 6) different combinations. Because these scenarios differ to a large degree, it is expected that the SVR displays a severe deterioration in performance. This is shown in table 10, where the MAPE of the Support Vector Regression model is given for time and cost forecasting. The diagonal in the table corresponds with those situations where the training set is equal to the test set, whereas the final column represents the average MAPE across the 6 test set scenarios. Compared to section 5.2 where the maximum MAPE was 3.22% for time forecasting and 1.98% for cost forecasting, the MAPE now reaches values of 20% on average. Based on table 10, the following conclusions can be drawn:

- The largest forecasting errors can be found for the scenarios with subscript 2. This finding is little surprising since these scenarios represent a higher degree of uncertainty. For time forecasting, the training set with the highest average MAPE is E_2, while L_2 is the worst performing scenario for cost forecasting.

- A larger discrepancy between training and test generally entails a higher MAPE. Typically, the forecasting accuracy for an early training set and a late test set will be worse compared to an on time test set. This trend is valid for time and cost forecasting.

It is hardly surprising that the SVR method does not perform well when the training set greatly differs from the test set. This corresponds with the real life situation where a project manager makes a wrong appraisal of the expected variability. Because the Artificial Intelligence approaches

	Training Set	Test Set						Average
		E_1	E_2	OT_1	OT_2	L_1	L_2	
Time	E_1	1.61%	16.43%	6.44%	11.28%	12.17	15.87%	10.63%
	E_2	18.13%	2.93%	22.5%	22.71%	27.22%	32.1%	20.93%
	OT_1	7.88%	13.74%	1.05%	8.28%	10.44%	15.5%	9.48%
	OT_2	13.95%	20.28%	13.86%	2.58%	15.53%	18.42%	14.10%
	L_1	15.95%	25.14%	10.09%	12.55%	1.43%	4.68%	11.64%
	L_2	24.66%	36.99%	16.29%	20.11%	6.45%	2.21%	17.79%
Cost	E_1	1.04%	11.1%	7.74%	8.18%	15.37%	18.21%	10.27%
	E_2	11.85%	1.97%	16.3%	16.24%	21.38%	26.52%	15.71%
	OT_1	8.04%	15.56%	0.69%	4.71%	7.71%	11.98%	8.12%
	OT_2	9.36%	20.28%	11.14%	1.68%	13.03%	17.31%	12.13%
	L_1	19.25%	24.71%	10.27%	9.67%	0.91%	4.82%	11.61%
	L_2	24.79%	34.47%	14.25%	19.58%	5.39%	1.44%	16.65%

Table 10: Overview of the SVR MAPE when the training set differs from the test set

rely on historical data, this section exemplifies the garbage in, garbage out principle found in other techniques (e.g. risk analysis). In the following section, we will study the situation where the training set is similar to the test set, without stemming from the same scenario (cf section 5.2) or from a totally different scenario. Incidentally, this situation, in which the project manager has some knowledge about the distribution without knowing the exact parameters, will paint a truer picture of the robustness capabilities of the proposed SVR method.

5.3.2. Training set $\approx$ Test set

In this section, the effect of a training set that is similar to the test set on the overall forecasting performance is evaluated. In order to do this, several distributions, based on the 6 scenarios proposed in section 4 were generated. The distributions can be partitioned into 3 different classes, as described along the following lines:

- Symmetric: using the three extreme scenarios (E_2, OT_2, L_2) or the 6 scenarios (E_1, E_2, OT_1, OT_2, L_1 and L_2), a symmetric generalized beta distribution is fitted. As a result, this fitted distribution takes the densities from the different distributions into account.
- Random: the distribution that specifies the activity duration is chosen randomly from the (sub)set of 6 scenarios. Hence, a random number is drawn for each activity, after which the duration uncertainty is drawn from the distribution belonging to the random number.
- Uniform: instead of a generalized beta distribution, a uniform distribution is used by specifying a lower bound a and an upper bound b.

Using these 3 classes, 6 additional scenarios were constructed. For the symmetric and random classes, the 3 extreme scenarios and all 6 scenarios were employed. For the uniform distribution, the lower (upper) bounds for the two distributions are equal to 0.5 (1.5) and 0.1 (1.9), respectively. The consequence is that these scenarios are similar to one another, but not identical. This allows us to inspect 36 combinations (6 * 6) resulting from the use of each scenario as a training and test set. In order to keep the analysis tractable, the results were averaged across all test sets and can be found in table 11 for time forecasting and in table 12 for cost forecasting. The standard deviation provides a measure of sensitivity and in this case represents the sensitivity due to the variation in test set. Once again, we have opted to report the MAPE of the best performing PV, ED, ES and Elshaer method. For cost predictions, the results from the 8 forecasting methods are reported. Obviously, these methods do not make use of historical data. Consequently, the MAPE does not differ based on a different training set.

Despite the fact that the training and test set are not identical, the proposed SVR method still performs very well. The SVR method yields better results in 5 out of the 6 training scenarios for the time dimension. The only scenario for which the performance is slightly worse is the uniform distribution with the lower bound equal to 0.1 and the upper bounds equal to 1.9. However, the slightly higher MAPE (4.44% versus ES's 4.12%) is offset by the lower value for σ (1.6 versus 2.24). For cost forecasting, the two best performing methods are SVR and EAC_1. The difference in performance compared to the other EAC methods is quite high. The SVR method's performance is on par with that of EAC_1, except for the same uniform distribution that yielded worse results for time forecasting. The deterioration is once again countered by the lower value for σ (1.19 versus 1.44 for EAC_1).

The goal of this section was to investigate the robustness of the proposed SVR method. This was done by varying the degree of resemblance between the training and test set. In the first section, the training set was completely different from the test set, leading to MAPE values of up to 20% on average. In the second section, the test set resembled the training set more closely, without being identical as was done in section 5.2. Our findings revealed that the forecasting performance in this case is better than or as good as the best performing traditional forecasting method. These observations strengthen our belief in the successful application of Support Vector Regression for project control forecasting, even when the training set deviates from the test set.

Training Set		SVR		PV		ED		ES		Elshaer	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Symmetric	3 Scenarios	3.92%	2.21	4.33%	2.31	4.37%	2.36	4.12%	2.24	7.21%	3.83
	6 Scenarios	3.95%	2.25								
Random	3 Scenarios	4.03%	2.28	4.33%	2.31	4.37%	2.36	4.12%	2.24	7.21%	3.83
	6 Scenarios	3.99%	2.34								
Uniform	0.5-1.5	3.93%	2.13	4.33%	2.31	4.37%	2.36	4.12%	2.24	7.21%	3.83
	0.1-1.9	4.44%	1.6								

Table 11: Overview of the SVR MAPE when the training set is similar to the test set (time)

Training Set		SVR		EAC_1		EAC_2		EAC_3		EAC_4		EAC_5		EAC_6		EAC_7		EAC_8	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Symmetric	3 Scenarios	2.57%	1.44	2.57%	1.44	4.64%	2.73	5.68%	3.2	5.55%	3.23	8.27%	4.52	8.15%	4.52	4.67%	2.73	4.64%	2.73
	6 Scenarios	2.58%	1.46																
Random	3 Scenarios	2.66%	1.53	2.57%	1.44	4.64%	2.73	5.68%	3.2	5.55%	3.23	8.27%	4.52	8.15%	4.52	4.67%	2.73	4.64%	2.73
	6 Scenarios	2.61%	1.54																
Uniform	0.5-1.5	2.61%	1.47	2.57%	1.44	4.64%	2.73	5.68%	3.2	5.55%	3.23	8.29%	4.52	8.15%	4.52	4.67%	2.73	4.64%	2.73
	0.1-1.9	2.78%	1.19																

Table 12: Overview of the SVR MAPE when the training set is similar to the test set (cost)

6. Conclusion

This paper presents a six-step methodology for applying a well-known Artificial Intelligence method, Support Vector Machines, to a project control setting. In the first phase, a diverse set of project networks was generated. Next, Monte Carlo simulations were used in order to introduce variability in the durations of individual activities. The resulting project control data was measured periodically and captured using Earned Value Management indicators. The data set was then partitioned into a training and test set. Since every method requires some parameter tuning, cross-validation using 5 folds and a grid search procedure were implemented. The parameter settings were then applied to a test set, which allowed us to compare the forecasting performance of the SVR to the current time and cost forecasting methods.

When the training set is identical to the test set, the results show that the SVR outperforms the other forecasting methods across all levels of the SP-factor and all Monte Carlo simulation settings, which were captured using 6 scenarios. The SVR model displays the characteristic improvement in time performance as the project progresses. Like many other methods, Support Vector Regression is a “garbage-in garbage-out” method, which requires the project manager to have a good understanding of the subsequent variability. This was demonstrated using an experiment in which all levels of the training and test set were combined. At the same time, we have shown that if the training set resembles the test set without being either identical or completely different, the superior performance of the SVR method can be maintained. While it is natural for learning methods to be subject to a greatly varying performance if the training set does not correspond with the test set, the robustness experiment reveals a caveat for project managers wishing to apply the proposed SVR model. If the project manager fails to correctly appraise the test set (or by extension, its parameters), it is wiser to opt for one of the currently available forecasting methods. However, if it is possible to realistically determine the parameters that give rise to an activity’s variability, significant forecasting improvements can be gained from the use of the SVR model. This statement holds true even if the parameters of the training and test set are similar in nature.

Our future research avenues are two-fold. The first direction follows from a limitation of the current manuscript. The results in this paper are based on simulations and have yet to pass the test of empirical validation. However, the topologically diverse data set should be regarded as as a trigger to apply the SVR model and incorporate sector-specific attributes, as was done in the work by Cheng et al. (2010). Secondly, we are convinced that the use of Support Vector Machines is only the proverbial tip of the iceberg. The field of Artificial Intelligence boasts many alternative methods that can be compared and contrasted. Furthermore, the application of learning methods using EVM data is not restricted to forecasting but can be extended to other project control problems that facilitate the timing and nature of taking corrective actions.

Acknowledgements

This work was carried out using the Stevin Supercomputer Infrastructure at Ghent University, funded by Ghent University, the Hercules Foundation and the Flemish Government - department EWI. Furthermore, we acknowledge the support given by the "Fonds voor Wetenschappelijk Onderzoek (FWO)" under contract number G/0095.10N.

References

AbouRizk, S., Halpin, D., and Wilson, J. (1994). Fitting beta distributions based on sample data. *Journal of Construction Engineering and Management*, 120:288–305.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

Cheng, M.-Y., Peng, H.-S., Wu, Y.-W., and Chen, T.-L. (2010). Estimate at completion for construction projects using evolutionary support vector machine inference model. *Automation in Construction*, 19(5):619–629.

Cheng, M.-Y. and Roy, A. F. (2010). Evolutionary fuzzy decision model for construction management using support vector machine. *Expert Systems with Applications*, 37(8):6061–6069.

Christensen, D. (1993). The estimate at completion problem: A review of three studies. *Project Management Journal*, 24:37–42.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Demeulemeester, E., Vanhoucke, M., and Herroelen, W. (2003). Rangen: A random network generator for activity-on-the-node networks. *Journal of Scheduling*, 6:17–38.

Elshaer, R. (2013). Impact of sensitivity information on the prediction of project's duration using earned schedule method. *International Journal of Project Management*, 31:579–588.

Fleming, Q. and Koppelman, J. (2003). What's your project's real price tag? *Harvard Business Review*, 81:20–21.

Fleming, Q. and Koppelman, J. (2005). *Earned value project management. 3rd Edition.* Newtown Square, PA: Project Management Institute.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A practical guide to support vector classification. `http://csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

Jacob, D. and Kane, M. (2004). Forecasting schedule completion using earned value metrics? Revisited. *The Measurable News*, Summer:1, 11–17.

Johnson, D. (1997). The triangular distribution as a proxy for the beta distribution in risk analysis. *The Statistician*, 46:387–398.

Keerthi, S. and Lin, C.-J. (2003). Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation*, 15(7):1667–1689.

Kuhl, M. E., Lada, E. K., Steiger, N. M., Wagner, M. A., and Wilson, J. R. (2007). Introduction to modeling and generating probabilistic input processes for simulation. In Henderson, S., Biller, B., Hsieh, M., Shortle, J., Tew, J., and Barton, R., editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 63–76. New Jersey: Institute of Electrical and Electronics Engineers.

Lipke, W. (2003). Schedule is different. *The Measurable News*, Summer:31–34.

Mangasarian, O. (2003). Data mining via support vector machines. In *System Modeling and Optimization XX*, pages 91–112. Springer.

Smola, A. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.

Uyttewael, E. (2005). *Dynamic Scheduling with Microsoft Office Project 2003: the book by and for professionals.* Co-published with International Institute for Learning Inc, Boca Raton.

Vandevoorde, S. and Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management*, 24:289–302.

Vanhoucke, M. (2010a). *Measuring Time - Improving Project Performance using Earned Value Management*, volume 136 of *International Series in Operations Research and Management Science.* Springer.

Vanhoucke, M. (2010b). Using activity sensitivity and network topology information to monitor project time performance. *Omega The International Journal of Management Science The International Journal of Management Science*, 38:359–370.

Vanhoucke, M. (2011). On the dynamic use of project performance and schedule risk information during project tracking. *Omega The International Journal of Management Science*, 39:416–426.

Vanhoucke, M. (2012). *Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control*, volume XVIII. Springer.

Vanhoucke, M. (2014). *Integrated Project Management and Control: First come the theory, then the practice.* Management for Professionals. Springer.

Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., and Tavares, L. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research*, 187:511–524.

Vanhoucke, M. and Vandevoorde, S. (2007). A simulation and evaluation of earned value metrics to forecast the project duration. *Journal of the Operational Research Society*, 58:1361–1374.

Vapnik, V. (1998). *Statistical learning theory.* John Wiley and Sons, New York.

Vapnik, V. (2000). *The nature of statistical learning theory.* Springer.

Appendix A. Example

In this section, the methodology will be explained using a toy example. It is worth noting that we assume that the training phase is complete and hence, the optimal parameters are found. Throughout this example, $C^* = 2^9$ and $\gamma^* = 2^{-15}$. Furthermore, only 1 fold will be used.

Network Generation. The toy example has an SP equal to 0.7 and can be thought of as one of the data instances of the test set of table 4. Consider the Activity on the Node (AoN) network given in figure A.1. Apart from the topological structure, it is necessary to determine the baseline durations and baseline costs. These are drawn from a random distribution between 20 and 40 and 50 and 100, respectively. The baseline duration is indicated above each node, while the variable cost per time unit is denoted underneath each node. Activities 1 and 12 are dummy activities and merely serve to indicate the project's start and end, respectively. The Planned Duration (PD) of the project is equal to 201 time units and the Budget At Completion (BAC) is equal to €21,619.

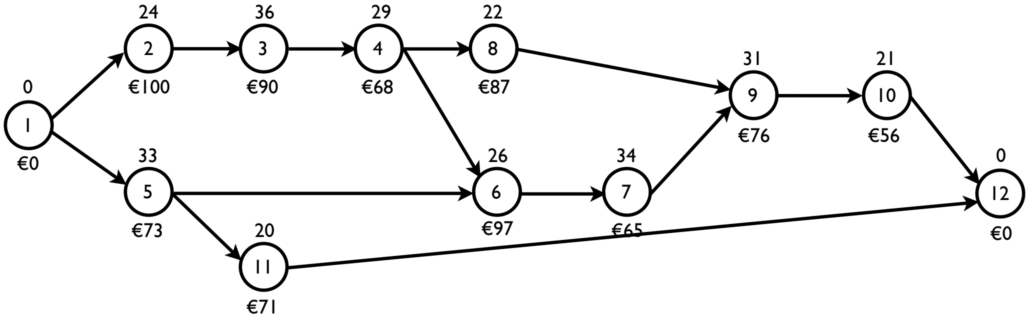


Figure A.1: Activity on the Node network of the example

Monte Carlo simulation. The Monte Carlo simulations allow to introduce time and cost variability on the activity level using a generalized beta distribution. In this example, the distribution of the L_1 scenario (cf table 3) was used. In order to keep the amount of data within limits, 10 executions were performed. For each execution, a draw from the generalized beta distribution for the duration of every activity is performed, leading to a real duration that deviates from the baseline duration. The real duration for every activity across the 10 executions are provided in table A.1. The executions will be partitioned into a learning set that consists of 8 executions (80%) and a test set of 2 executions (20%). These percentages are identical to those of table 4.

Attributes. The attributes are the inputs that are used by the Support Vector Machine model to learn their relation to the output measures. In this example, we aim to predict the Real Duration of the two executions of the test set. For duration forecasting, 19 attributes (SPI, SPI(t), CPI, ES, 9 forecasting methods and 6 Elshaer forecasting methods) for every percentage complete can be

Ex	Activity									
	1	2	3	4	5	6	7	8	9	10
1	29	42	27	31	26	33	21	29	21	24
2	24	38	29	36	24	39	24	36	24	20
3	30	36	32	36	32	44	25	30	21	20
4	26	35	27	39	24	34	24	30	21	20
5	25	48	30	39	32	39	21	31	23	23
6	23	38	40	32	32	36	24	29	24	22
7	31	35	30	32	27	32	20	30	25	21
8	26	37	34	31	27	38	29	31	20	21
9	24	41	34	32	28	34	23	31	24	22
10	27	40	29	37	30	35	26	30	26	19

Table A.1: Real Duration of every activity for the 10 Monte Carlo executions

used (cf table 2). In order to keep the amount of data small, we will only work with data at the 10% complete point. An overview of the data is given in table A.2, where the first 8 executions constitute the learning set (RD is known) and the final 2 executions are runs for which the project is 10% complete and an estimate of the final duration is required.

Cross-validation and Grid Search. For this toy example, it is assumed that only 1 fold is used and that the grid search (the training set part of table 4) has been completed. The optimal parameters are as follows: $C^* = 2^9$ and $\gamma^* = 2^{-15}$.

Testing. Based on the data of table A.2, a forecast for the traditional methods as well as the Support Vector Regression can be constructed. For instance, for ES_1, the forecast of execution 9 is equal to:

$$ES_1 = AD + (PD - ES) = 12 + 201 - 11.74 = 200.26 \tag{A.1}$$

Since we are only at the start of the project, T is equal to 1 in equation (8), which yields the following MAPE for execution 9:

$$MAPE_{ES_1} = \frac{|a_i - EAC(t)_1|}{a_i} = \frac{200.26 - 301}{301} = 0.3347 \tag{A.2}$$

The working of the Support Vector Machine is detailed in figure A.2. In the first phase, the learning set is used to train the SVM. It learns the relation between the input attributes of the first 8 executions (learning set) and the real duration. After this relation has been captured, it can be applied to new data. These data, executions 9 and 10, make up the test set. Obviously, at the 10% complete point, the real duration is not known, which means a prediction needs to be made. The

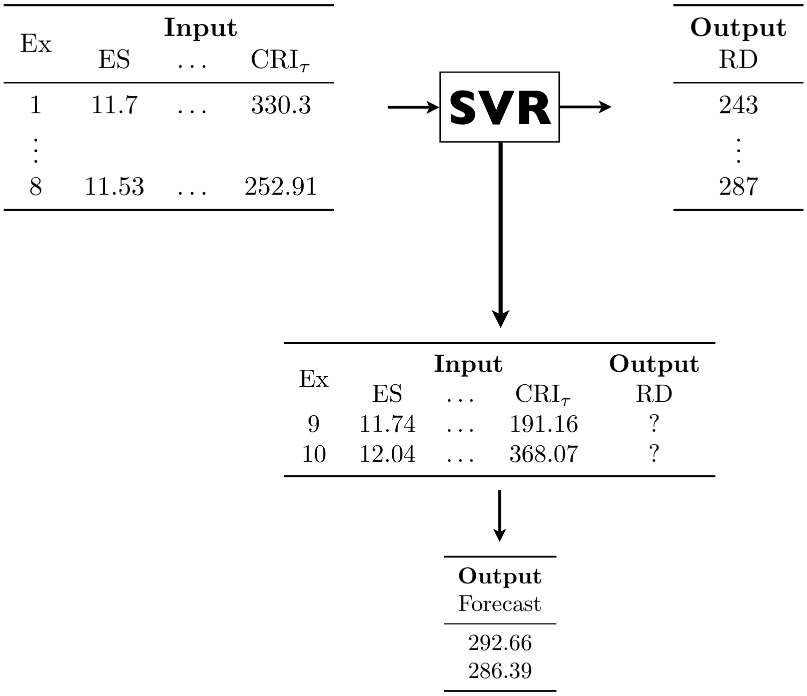
SVR employs the relation it learned from the learning set and the input attributes of the test set to construct a forecast. Hence, the forecast is constructed using the following data and settings, where A.2 refers to table A.2:

$$SVR.Relation = SVR(RD \sim ., train = A.2_{row1-8}, kernel = radial, C = 2^{9}, \gamma = 2^{-15}) \tag{A.3}$$

After the relation (denoted using the tilde, ∼) between the Real Duration (RD) and all the input attributes has been learnt (denoted in equation (A.3) by the full stop, .), a forecast can be constructed using the relation and the attributes of the test data:

$$SVR.Forecast = predict(SVR.Relation, test = A.2_{row9}) \tag{A.4}$$

Applying equation (A.4) to execution 9 then yields a forecast value of 292.66. The forecast values, as well as their MAPE, can be found in table A.3. Even though this example is not representative to make general conclusions, the mean MAPE (averaged across the two executions) of the SVR model is much lower than the incumbent method, the Elshaer method with the Criticality Index.



Ex	**Input**		
	ES	…	CRI_{τ}
1	11.7	…	330.3
⋮			
8	11.53	…	252.91

Output
RD
243
⋮
287

Ex	**Input**			**Output**
	ES	…	CRI_{τ}	RD
9	11.74	…	191.16	?
10	12.04	…	368.07	?

Output
Forecast
292.66
286.39

Figure A.2: Detailed overview of the working of the Support Vector Machine

Ex	RD	ES	SPI	CPI	SPI(t)								Forecasts								
																	Elshaer			
						PV$_1$	PV$_2$	PV$_3$	ED$_1$	ED$_2$	ED$_3$	ES$_1$	ES$_2$	ES$_3$	CI	SI	SSI	CRI$_r$	CRI$_\rho$	CRI$_\tau$
1	243	11.7	0.9	0.9	0.9	203.1	223.4	248.3	202.3	223.4	246.85	202.3	223.4	246.85	424.3	288.63	420.78	396.58	382.39	330.3
2	288	12.02	0.8	0.8	0.8	205.79	250.82	313	203.98	250.82	309.28	203.98	250.82	309.28	218.55	234.31	218.81	220.76	222.03	227.83
3	339	12.37	0.52	0.52	0.52	219.7	389.9	756.33	212.63	389.9	733.78	212.63	389.9	733.78	466.37	423.01	465.52	459.37	455.48	439.12
4	196	12.33	0.56	0.56	0.56	216.55	358.61	639.82	210.67	358.61	622.57	210.67	358.61	622.57	327.44	343.02	327.7	329.67	330.94	336.7
5	365	12.44	0.62	0.62	0.62	213.15	323.05	519.2	208.56	323.05	507.06	208.56	323.05	507.06	277.49	299.57	277.85	280.57	282.34	290.45
6	330	12.41	1.13	1.13	1.13	198.74	178.21	158	199.59	178.21	159.25	199.59	178.21	159.25	184.08	180.96	184.02	183.61	183.34	182.18
7	267	12.49	0.78	0.78	0.78	206.64	257.44	329.74	204.51	257.44	325.25	204.51	257.44	325.25	504.97	336.08	500.5	469.85	452.01	387.17
8	287	11.53	1.05	1.05	1.05	200.14	191.71	182.85	200.47	191.71	183.36	200.47	191.71	183.36	296.78	230.7	295.27	284.65	278.2	252.91
9	301	11.74	1.07	1.07	1.07	199.81	188.32	176.44	200.26	188.32	177.14	200.26	188.32	177.14	192.5	190.29	192.46	192.17	191.98	191.16
10	317	12.04	0.55	0.55	0.55	217.02	367.21	670.86	210.96	367.21	652.67	210.96	367.21	652.67	368.46	367.81	368.45	368.37	368.31	368.07

Table A.2: Sample learning and test set data at 10% complete

Ex	Metric	SVR	PV$_1$	PV$_2$	PV$_3$	ED$_1$	ED$_2$	ED$_3$	ES$_1$	ES$_2$	ES$_3$	CI	SI	Elshaer			
														SSI	CRI$_r$	CRI$_\rho$	CRI$_\tau$
9	Forecast	292.66	199.81	188.32	176.44	200.26	188.32	177.14	200.26	188.32	177.14	192.5	190.29	192.46	192.17	191.98	191.16
	MAPE	2.77%	33.62%	37.43%	41.38%	33.47%	37.43%	41.15%	33.47%	37.43%	41.15%	36.05%	36.78%	36.06%	36.16%	36.22%	36.49%
10	Forecast	286.39	217.02	367.21	670.86	210.96	367.21	652.67	210.96	367.21	652.67	368.46	367.81	368.45	368.37	368.31	368.07
	MAPE	9.66%	31.54%	15.84%	111.63%	33.45%	15.84%	105.89%	33.45%	15.84%	105.89%	16.23%	16.03%	16.23%	16.2%	16.19%	16.11%
Mean MAPE		6.21%	32.58%	26.64%	76.50%	33.46%	26.64%	73.52%	33.46%	26.64%	73.52%	26.14%	26.40%	26.14%	26.18%	26.20%	26.30%

Table A.3: Forecast for the sample data