

CASH FLOW PREDICTION FOR CONSTRUCTION PROJECT USING A NOVEL ADAPTIVE TIME-DEPENDENT LEAST SQUARES SUPPORT VECTOR MACHINE INFERENCE MODEL

Min-Yuan CHENG^a, Nhat-Duc HOANG^b, Yu-Wei WU^a

^a*Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taiwan*

^b*Institute of Research and Development, Duy Tan University, Vietnam*

Received 09 Sep 2012; accepted 08 Jan 2013

Abstract. Cash flow information is crucial for the decision making process in construction management. Due to the complexity and the dynamic progress of a construction project, forecasting cash flow demand throughout various phases of the project remains a challenging problem. This article presents a novel inference model, named as Adaptive Time-dependent Least Squares Support Vector Machine (LS-SVM_{AT}) for cash flow prediction. In the LS-SVM_{AT}, Least Squares Support Vector Machine (LS-SVM) is integrated with an adaptive time function (ATF) to generalize the input-output mapping of cash flow. Since cash flow data are time-dependent, data points recorded in different periods can contribute dissimilarly to the training process of the prediction model. Thus, the role of the ATF is to determine the appropriate weight associated with each data point at a specific time period. By doing so, LS-SVM_{AT} can better deal with the dynamic nature of the time series. Furthermore, to identify the optimal parameters for the inference model, Differential Evolution (DE) based cross validation process is utilized in this research. Comparing to other benchmark methods, the proposed model has identified the most appropriate time function and has yielded superior forecasting results. Therefore, LS-SVM_{AT} can be a promising tool for construction managers in cash flow prediction.

Keywords: cash flow prediction, least squares support vector machine, adaptive time function, Differential Evolution, artificial intelligence, construction management.

Introduction

Construction project is shown to be context-dependent and highly uncertain; this explains why the construction industry suffers the largest number of bankruptcies compared to other sectors of the economy (Boussabaine, Kaka 1998). The nature of construction projects, which is characterized by constant changes in the environment, pressures to maintain schedules and reduce costs with increasingly complex construction techniques, makes the task of project management a significant challenge.

In project management, cash is a critical factor that imposes significant influence on project profitability (Hwee, Tiong 2002; Jiang *et al.* 2011). Poor cash flow control can lead to project failure for contractors due to liquidity shortage for supporting their daily activities (Khosrowshahi, Kaka 2007). Hence, reliable cash flow prediction over various phases of a project is desirable since it puts the project manager in a better position to identify potential financial problems and to develop appropriate strategies to mitigate the negative effects of such problems on the project success (Hwang, Liu 2005).

For the purpose of project control, Russell *et al.* (1997) pointed out that one may identify and keep track of several time-dependent variables that change through the construction progress. Such method can help manager monitor the project status and foresee some undesirable events that may happen in the future. Nevertheless, predicting project performance dynamically in terms of cash flow is enormously challenging. It is because each time point is associated with numerous time-dependent variables.

Hence, another alternative for project control in terms of cash flow is to analyse the pattern of cash demand in the past and then infer that pattern into the future. This, in essence, is the time series forecasting approach that can effectively assist both short-term and long-term decision making (Williams 1994; Nam *et al.* 2007). Due to its practical importance, various approaches that employ traditional statistical methods as well as advanced artificial intelligence (AI) methods have been developed to tackle the time series problems.

A survey carried out by Sapankevych and Sankar (2009) found that real-world time series are too complex to be modelled using traditional statistical methods. Thus,

this demands more advanced forecasting algorithms such as Kalman filters, artificial neural networks (ANN), and support vector machines (SVM). The survey also pointed out that among AI methods, SVM has the ability to accurately forecast time series data especially when the data of interest are non-linear and non-stationary.

Moreover, SVM has been utilized in a variety of applications in construction engineering (An *et al.* 2007; Cheng *et al.* 2010). Hence, this method can be promising for cash flow problem. The principles of SVM are based on the structural risk minimization and statistical learning theory. After being trained, SVM is capable of predicting the future value of cash demand. The advantages of this method are strong inference capacity and generalization. Nevertheless, SVM training requires solving a quadratic programming problem subjected to inequality constraint. This means that the training process for large data sets requires expensive computational cost (Guo, Bai 2009).

LS-SVM has been proposed recently to overcome the drawback of SVM (Suykens *et al.* 2002; De Brabanter *et al.* 2010). One obvious advantage of LS-SVM is the alleviation of the computational cost. In its training process, a least squares cost function is proposed to obtain a linear set of equations in the dual space. Consequently, to derive the solution, it is required to solve a set of linear equations, instead of the quadratic programming as in SVM. And, this linear system can be efficiently solved by iterative methods such as conjugate gradient. Studies have been carried out to demonstrate the prediction capability of LS-SVM (Yu *et al.* 2009; Samui *et al.* 2011).

However, one shortcoming of LS-SVM is that it does not consider the unbalanced feature of real-world time series. In construction industry, data related to cost are collected periodically in various phases of the project; thus, many factors of interest in construction lend themselves to time series behaviour (Khosrowshahi, Kaka 2007). It is obvious that during the project execution, data recorded more recently could provide more information for the decision makers. It is because cost items reported in distant past may be less relevant due to changes in the working environment. In cash flow prediction, data collected closer to the timing of prediction should be more important than that which is more distant. Furthermore, the level of importance of data recorded in different phases of the project may not be similar. Thus, any forecasting techniques applied for the case of cash flow should take into account this very nature of time series.

To address the issue of unbalanced learning, Lin and Wang (2002) first proposed the weight, which is expressed in terms of a fuzzy membership, associated with each input point; and this allows each one to contribute dissimilarly in the training process. Two time functions, namely linear and quadratic, are used to calculate those weights. Nevertheless, since each time series and each domain problem may possess different characteris-

tics, utilizing deterministic types of time function may not yield desirable performance. Thus, this article aims to put forward a flexible form of time function, namely ATF, which is capable of adapting itself to different time series problems in an autonomous manner.

Additionally, in AI field, it is recognizable that tuning parameters play an important role in establishing the predictive model (Suykens 1999; Bishop 2006). These parameters control the model's complexity, and they are needed to be determined properly via cross-validation. In doing so, the main objective is to obtain an optimal model that is capable of producing the best predictive performance on new data. In this study, DE, a fast and effective stochastic optimizer proposed by Storn and Price (Price *et al.* 2005), is employed in the cross-validation process to achieve such objective.

Therefore, this article aims to propose a hybrid AI model that employs various advanced techniques to help project managers in predicting the future cash demand. The second section of this paper describes the proposed approach LS-SVM_{AT}. The cross validation process for optimizing model's parameters is mentioned in the third section. The fourth section demonstrates application of the inference model in cash flow forecasting. Conclusion on our study is mentioned in the final section.

1. Adaptive time-dependent least squares support vector machine

LS-SVM is a supervised learning technique proposed for solving both classification and regression problems. For standard formulation of LS-SVM, readers are guided to previous works of Suykens *et al.* (2002) and De Brabanter *et al.* (2010). This section aims to describe the LS-SVM_{AT} which incorporates LS-SVM with an adaptive form of time function.

Consider the following regression model that defines the mapping relationship between a response variable and independent variables:

$$y(x) = w^T \phi(x) + b, \quad (1)$$

where: $x \in R^n$, $y \in R$, and $\phi(x) : R^n \rightarrow R^{nh}$ is the mapping to the high dimensional feature space.

Given a training dataset $\{x_k, y_k\}_{k=1}^N$, the formulation of LS-SVM_{AT} can be given as follows:

$$\text{Minimize } J_p(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N s_k e_k^2, \quad (2)$$

Subjected to $y_k = w^T \phi(x_k) + b + e_k$, $k = 1, \dots, N$, where: $w \in R^n$ is the normal vector to the regression hyperplane; $b \in R$ is the bias; $e_k \in R$ is an error variable; $\gamma > 0$ denotes a regularization constant; $s_k \in [0, 1]$ is a time-dependent weight associated with an error variable.

It is noted that s_k is a function of time $s_k = f(t_k)$ where t_k is the time period of the data point k . As

mentioned earlier, real-world time series data can be unbalanced due to the fact that recent data can be more relevant than distant ones. Therefore, data points should be weighted differently according to time. This study introduces the ATF for determine the value of s_k used in Eqn (2). Consider the case of a construction project, the total project duration can be divided into a number of completion periods (U). Thus, the proposed time function assigns small weighting values for data points at the initial phase of the project. Meanwhile, data points recorded at the later phase are coupled with greater weights because they possess more valuable information for the training process (Fig. 1). Using the proposed ATF, the duration of a completed project is divided into T time domains. It is noted that the number of time domains (T) is not necessarily equal to the number of completion periods (U).

For the first time domain D_1 , the time function has two free parameters: the initial value and the slope. The weights within D_1 are calculated as following:

$$s_k = s_o + a_1(t_k), \quad (3)$$

where: s_k denotes the weighting value associated with the data point k ; t_k is the time period; s_o is the initial value of the weighting value and it is usually set to be relatively close to 0; a_1 represent the slope parameter.

Meanwhile, the time functions for other domains only need the slope parameter to specify its shape. The weights within a time domain D_v are calculated in the following way:

$$s_k = s_{o,v-1} + a_v(t_k), \quad (4)$$

where: $s_{o,v-1}$ is the maximum weighting value in the previous time period; a_v represents the slope parameter.

To solve the optimization problem stated in Eqn (2), one can construct the Lagrangian and derive the dual problem. Similar to original LS-SVM (Suykens *et al.* 2002), the Lagrangian is given by:

$$L(w, b, e; \alpha) = J_p(w, e) - \sum_{k=1}^N \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\}, \quad (5)$$

where α_k are Lagrange multipliers.

And the conditions for optimality are given by:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^N \alpha_k \phi(x_k) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^N \alpha_k = 0 \\ \frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma s_i e_k, \quad k = 1, \dots, N \\ \frac{\partial L}{\partial \alpha_k} = 0 \rightarrow w^T \phi(x_k) + b + e_k - y_k = 0, \quad k = 1, \dots, N. \end{cases} \quad (6)$$

After elimination of e and w , the following linear system is obtained:

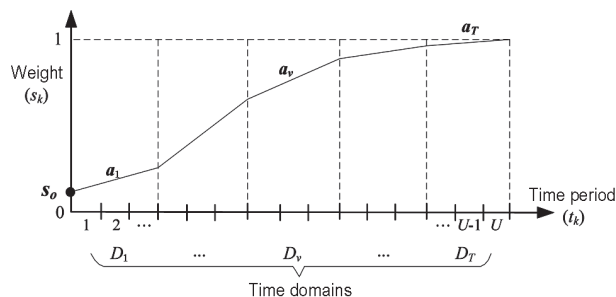


Fig. 1. Adaptive time function (ATF)

$$\begin{bmatrix} 0 & -Y^T \\ Y & \omega \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (7)$$

where:

$$Y = (y_1, y_2, \dots, y_N)^T; \quad (8)$$

$$\alpha = (\alpha_1; \dots; \alpha_N); \quad (9)$$

$$\omega_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) + (s_i \gamma)^{-1} I; \quad (10)$$

$$1 = (1, 1, \dots, 1)^T. \quad (11)$$

And the kernel function is applied as follows:

$$K(x_k, x_l) = \phi(x_k)^T \phi(x_l). \quad (12)$$

The resulting LS-LSVM_{AT} model for function estimation is expressed as:

$$y(x) = \sum_{k=1}^N \alpha_k K(x_k, x_l) + b, \quad (13)$$

where α_k and b are the solution to the linear system (7). The kernel function that is often utilized is Radial Basis Function (RBF) kernel given as following:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l\|^2}{2\sigma^2}\right), \quad (14)$$

where σ denotes the kernel function parameter.

It is noticed that the regularization constant (γ), kernel function parameter (σ), the initial weight (s_o), and the slope parameters (a_i) are the tuning parameters of the proposed LS-LSVM_{AT}. In our research, these tuning parameters are automatically optimized by the DE based cross validation process which is demonstrated in the next section of the article.

2. Optimizing model tuning parameters

2.1. Differential Evolution

Differential Evolution (DE) is an Evolutionary Algorithm which is designed for real parameter optimization (Price *et al.* 2005). DE algorithm relies on the implementation of a novel crossover-mutation operator, based on the linear combination of three different individuals

and one subject-to-replacement parent (or target vector) (Becerra, Coello 2006). The crossover-mutation operator yields a trial vector (or child vector) which will compete with its parent in the selection operator. The selection process is performed via selection between the parent and the corresponding offspring (Mezura-Montes *et al.* 2004). The algorithm of DE is depicted in Figure 2. In this figure, it is noted that NP represents the size of the population; $X_{j,i}$ is the j^{th} decision variable of the i^{th} individual in the population; g is the current generation; and D denotes the number of decision variables. $rand_j(0,1)$ is a uniform random number lying between 0 and 1; and $rnb(i)$ is a randomly chosen index ranging between 1 and NP .

In the selection process, the trial vector is compared to the target vector (or the parent) (Storn, Price 1997). If the trial vector can yield a lower objective function value than its parent, then the trial vector replaces the target vector. The selection operator is expressed as following:

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{if } f(U_{i,g}) \leq f(X_{i,g}) \\ X_{i,g} & \text{if } f(U_{i,g}) > f(X_{i,g}) \end{cases}, \quad (15)$$

where: $X_{i,g}$ represents the parent vector at generation g ; $U_{i,g}$ denotes the trial vector at generation g ; $X_{i,g+1}$ is the chosen individual which survives to the next generation ($g+1$).

The optimization process iterates until the stopping criterion is satisfied. The user can set the type of this stopping condition. Commonly, maximum generation (G_{\max}) or maximum number of function evaluations (NFE) can be applied as the stopping condition. When the optimization process terminates, the final optimal solution is available for the user assessment.

2.2. Differential Evolution based cross validation

In machine learning, one important goal is to construct a prediction model that can deliver the best generalization. It is because model performance on the training data set

is not necessarily a good indicator of the predictive performance on testing data due to the problem of over-fitting (Bishop 2006). Over-fitting arises when a regression model fits the training set very well, but performs poorly on the new data set. Hence, to build a desirable prediction model, one commonly used technique is the S -fold cross-validation (Suykens *et al.* 2002; Bishop 2006; Samarasinghe 2006). Employing this approach, the training data is divided into S folds and this allows a proportion $(S-1)/S$ of the available data to be used for training while other portion of the data is for assessing model performance.

Since our study employs LS-SVM_{AT} as the regression machine, its tuning parameters consist of the regularization parameter γ , the RBF kernel parameter σ , the initial weight s_o , and the slope parameters a_v . The proposed cross validation approach utilizes DE (Price *et al.* 2005) to automatically explore various combinations of (γ , σ , s_o , and a_v) and to identify the optimal set of these tuning parameters. In the following section, the DE-based cross-validation (Fig. 3) is described in detail.

In the step of data processing, the training data set is divided into S folds, in our study S is selected to be five. In each run, one fold is used as a validating set; meanwhile, the other folds are used for training the model. In LS-SVM_{AT} training, the machine is utilized to learn the mapping function between input and output for each run. After the training process, the propose method is applied to predict the output of the validating sets. In order to determine the optimal tuning parameters, the following objective function is used in the step of fitness function evaluation:

$$F_{fitness} = \frac{\sum_{k=1}^5 E_{tr}^k}{5} + \frac{\sum_{k=1}^5 E_{va}^k}{5}, \quad (16)$$

where E_{tr}^k and E_{va}^k denote the training and validating error, respectively, for the k^{th} run. The training and validat-

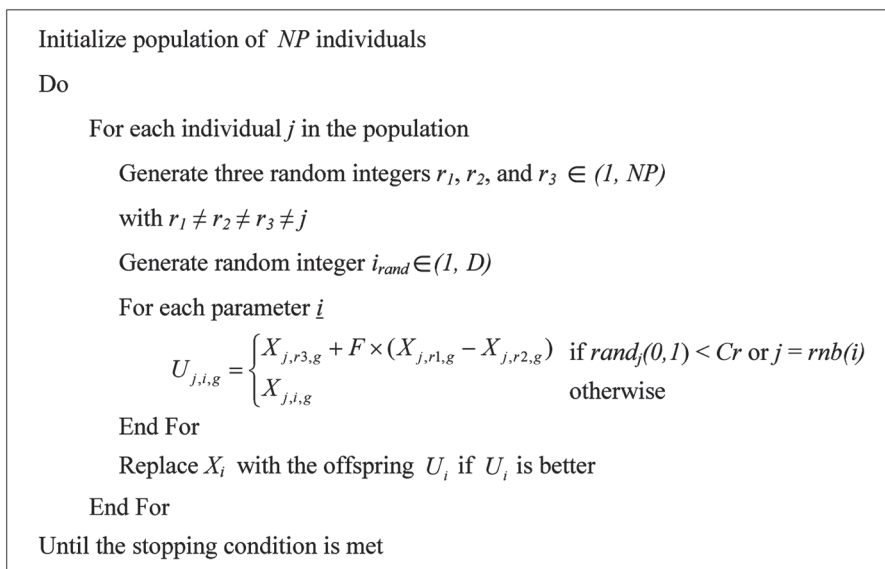


Fig. 2. Differential Evolution optimization algorithm

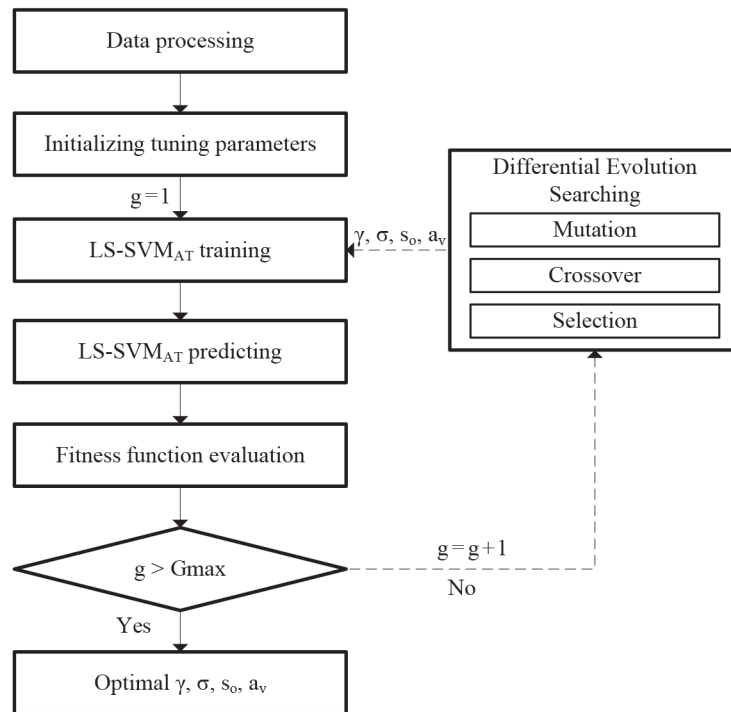


Fig. 3. Differential Evolution based cross-validation process

ing errors herein are Root Mean Squared Error (RMSE) calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^N (Y_P^j - Y_A^j)^2}{N}}, \quad (17)$$

where Y_P^j and Y_A^j represent predicted and actual value for output j^{th} . In addition, N represents the number of training data points used in each run.

The fitness function, in essence, represents the trade-off between model generalization and model complexity. It is worth noticing that well-fitting of the training set may reflect the model complexity. However, complex model tends to suffer from over-fitting (Suykens *et al.* 2002; Bishop 2006). Thus, incorporating the error of the validating data can help identify the model that features the balance between minimizing training error and achieving generalization capability.

In each generation, the DE algorithm carries out mutation, crossover, and selection process to guide the initial population to the final optimal solution. The search terminates when the current generation g achieves the maximum number of generation G_{max} . After being optimized, the prediction model is ready to be used in the next step.

3. LS-SVM_{AT} for cash flow prediction

3.1. Experimental results

This section of the article illustrates the performance of the proposed inference model LS-SVM_{AT} in real-world cash flow prediction problems. The database used in the paper was collected from a construction contractor in Taipei from

1996 to 2006. Herein, the standard cumulative cost-time curves were employed for cash flow forecasting. As cash flow is recorded sequentially, it features characteristics of time series. Hence, LS-SVM_{AT}, which fuses LS-SVM and ATF, can be very potential solve to the problem at hand.

The employed database contains percentage of expenditure cash flow taken from 13 high rise building projects. The LS-SVM_{AT} utilizes 10 projects as training set and 3 projects as testing set. Every project was separated into 20 sections; each section represents a uniform period of 5% total project completion. It is noted that for weighting data, the total project duration is divided into 5 time domains ($T = 5$).

To predict future cash flow demand, three sequential periods of expenditure cash flow were utilized as input patterns. There are 17 input data in a completed project from the first set (1, 2, 3) to the final set (17, 18, 19). Prediction results are represented by the cumulated cash flow ratio of the 4th through the 20th periods. Hence, 170 data points are employed to train the inference model and 51 data cases for testing its performance. Table 1 illustrates the cash flow data for one project in the database. In Table 1, the three inputs X_1 , X_2 , and X_3 represent the pattern of expenditure cash flow in the past; meanwhile, the output Y denotes the forecasted cash demand of the project.

To illustrate that LS-SVM_{AT} is capable of delivering accurate predictive results, the proposed model is benchmarked with Evolutionary Support Vector Machine Inference Model (ESIM) (Cheng, Wu 2009). ESIM is established based on the standard SVM without any weighting mechanism. Additionally, LS-SVM models that utilizes linear and quadratic time functions proposed by Lin and Wang

Table 1. Expenditure cash flow for one construction project

Case	Input pattern			Output
	X ₁	X ₂	X ₃	Y
1	0.34	4.97	6.38	7.21
2	4.97	6.38	7.21	9.71
3	6.38	7.21	9.71	15.99
4	7.21	9.71	15.99	23.45
5	9.71	15.99	23.45	27.55
6	15.99	23.45	27.55	31.74
7	23.45	27.55	31.74	36.25
8	27.55	31.74	36.25	42.20
9	31.74	36.25	42.20	47.69
10	36.25	42.20	47.69	54.04
11	42.20	47.69	54.04	58.72
12	47.69	54.04	58.72	72.45
13	54.04	58.72	72.45	78.16
14	58.72	72.45	78.16	82.04
15	72.45	78.16	82.04	90.65
16	78.16	82.04	90.65	95.23
17	82.04	90.65	95.23	100.00

(Lin, Wang 2002) are also used for result comparison; the two models are named as LS-SVM_L and LS-SVM_Q. The formulations of the linear and the quadratic function are shown in Eqn (18) and Eqn (19), respectively:

$$s_k = f_l(t_k) = \frac{1-\sigma}{t_m-t_1} t_i + \frac{t_m\sigma-t_1}{t_m-t_1}; \tag{18}$$

$$s_k = f_q(t_k) = (1-\sigma)\left(\frac{t_i-t_1}{t_m-t_1}\right)^2 + \sigma, \tag{19}$$

where: t_1 and t_m denote the first and the last time period of a project, respectively; σ represents the time function parameter which is the lower bound of weighting values. It is worth noticing that when LS-SVM is integrated with the two aforementioned time functions, all of model tuning parameters, including the regularization parameter, the kernel function parameter, and the time function parameter, are selected via the DE based cross validation process.

Moreover, to quantify the modelling accuracy of each approach, Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) have been employed as evaluation criteria. RMSE (Eqn (17)) is first computed by summing the squared deviations between the predicted and the observed outputs, then taking the square root of summation. Since the errors are squared before calculating the average deviation, more weights are given to data points with larger prediction errors. Thus, RMSE is effective for identifying undesirable large deviations. Meanwhile, MAPE quantifies the modelling performance by computing the ratio between the deviation and actual output. MAPE magnitude is not affected

by the unit of the predicted output; hence, it can express the difference between the actual and the predicted cash flow. MAPE is calculated as follows:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i^P - Y_i^A|}{Y_i^A} \times 100, \tag{20}$$

where: Y_i^A and Y_i^P are actual and predicted value for output i , respectively; n denotes the number of data points.

When the training process terminates, LS-SVM_{AT} as well as other benchmark models, including LS-SVM_L, LS-SVM_Q, and ESIM can be employed for predicting new cases of project cash flow. It is noticed that since the first three models utilized time functions for dealing with unbalanced learning, the optimal ATF for LS-SVM_{AT} is illustrated in Figure 4; meanwhile, the optimal shapes of linear and quadratic time function for LS-SVM_L and LS-SVM_Q are shown in Figure 5 and Figure 6, respectively.

The result comparison is provided in Table 2. It is recognizable that the proposed model has achieved the most desirable outcome in both training and testing set. For training set, RMSE and MAPE of LS-SVM_{AT} are 0.020 and 4.540, respectively. When predicting testing cases, LS-SVM_{AT} obtains 0.022 for RMSE and 4.731 for MAPE. Additionally, since the training error and testing error of LS-SVM_{AT} are relatively close to each other, it can be shown that the inference model does not suffer from over-fitting. This means DE based cross validation process is really effective to achieve a balance between model generalization and model complexity.

Furthermore, prediction results of the other two LS-SVM models, which also utilizes time functions, are better than that of ESIM, which is not equipped with time function. Thus, from the conducted experiments, it can be concluded that time functions are able to enhance the prediction capability. In addition, because forecasting outcome of LS-SVM_L is slightly better than that of LS-SVM_Q, it can be seen that model performance may vary when different types of time function are employed.

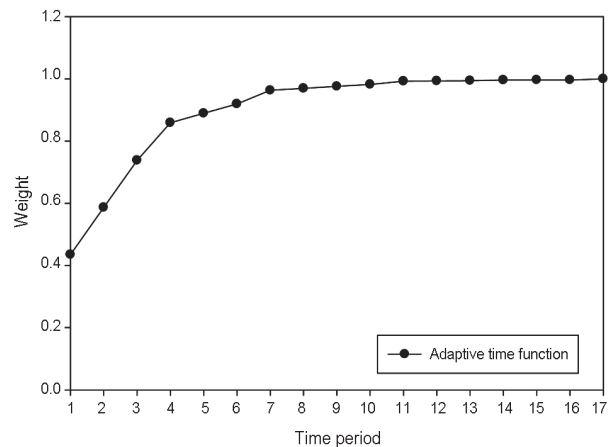


Fig. 4. Adaptive time function

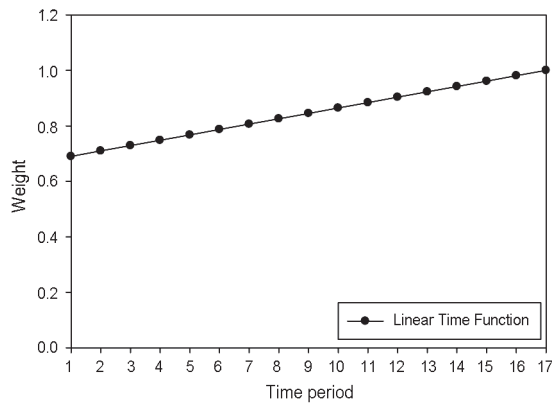


Fig. 5. Linear time function

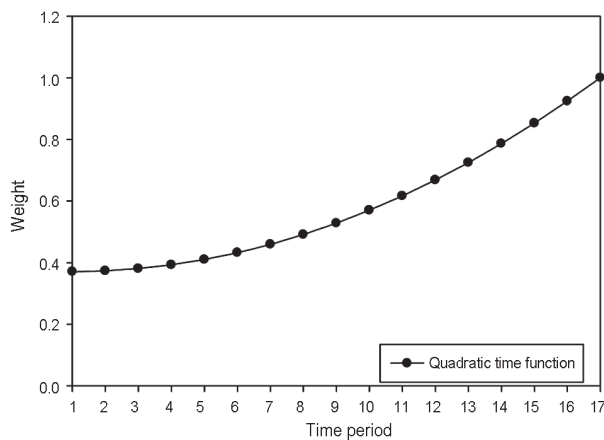


Fig. 6. Quadratic time function

The detailed results for 51 testing cases, including actual and predicted cash flow, are provided in Table 3. Herein, the result deviation is calculated as the absolute different between actual and predicted cash flow at a time period. The average deviation of LS-SVM_{AT}, LS-SVM_L, LS-SVM_Q, and ESIM are 1.83%, 2.30%, 2.51%, and 3.58%, respectively. Furthermore, the maximum deviations of the four models are 4.85%, 10.32%, 14.34%, and 8.98%. Hence, prediction performance in terms of result deviation of LS-SVM_{AT} is significant better than that obtained from other approaches. If we set 5% as the error threshold for predicting project cash flow, only the proposed inference model is qualified.

3.2. Discussion

Based on the results shown above, it can be seen that the LS-SVM_{AT}, which is fused with the newly developed

ATF, provides the best forecasting outcome. Thus, this form of time function, illustrated in Figure 4, is deemed best suited for the cash flow prediction problem. The ATF seems to be divided into two regions that are significantly different from each other. In the first region, from time period 1st to 7th, the weighting values increases gradually from 0.44 to 0.96. In the second region, the values of time function get stable reflected in an apparently straight line. Additionally, it can also be observed that the optimal time function is found to be concave-down. Meanwhile, the quadratic time function (Fig. 6), due the restriction of its functional formula, is always depicted in a concave-up shape. Thus, this explains why the quadratic function cannot capture the trend similar to that of the proposed adaptive method.

Furthermore, this weighting strategy of the ATF is understandable in the context a construction project. As in the initial stage of a project, fewer tasks were commenced and thus only a small proportion of works was carried out. The works on site at this stage are mostly preparatory activities; financial transactions and working information were less likely to appear. Thus, the weighting value for data points collected in this period should be low. When the project moves forward, it reaches a stable state in which most of the main activities are executed. Therefore, information collected at the later time periods should be considered more valuable and given greater weights. Accordingly, the ATF function is found to best demonstrate this feature as it delivers the best forecasting results.

Moreover, since LS-SVM_{AT} is a hybrid AI model, the approach can be quiet complex for practical managers. Nevertheless, construction management is a complicated field and predicting project performance is by no means an easy task. Therefore, it is very challenging to construct a simple model that yields highly accurate forecasting performance. Even though the proposed model is relatively complicated to establish, with more effort on software engineering, a user-friendly interface can be integrated into the model; and this enables LS-SVM_{AT} to be a more promising tool for practical project management.

Conclusions

This article has presented a novel approach, named as LS-SVM_{AT}, to assist construction managers in dealing with project cash flow forecasting. The proposed approach, which fuses LS-SVM and ATF, is developed as an intelligence model specifically designed for time

Table 2. Result comparison

Performance measurement		LS-SVM _{AT}	LS-SVM _L	LS-SVM _Q	ESIM
Training	RMSE	0.020	0.025	0.029	0.044
	MAPE	4.540	6.374	7.860	13.221
Testing	RMSE	0.022	0.030	0.035	0.043
	MAPE	4.731	5.778	6.148	13.899

Table 3. Detailed result of testing cases

Case	Actual output	LS-SVM _{AT}		LS-SVM _L		LS-SVM _Q		ESIM	
		Pred.	Dev.	Pred.	Dev.	Pred.	Dev.	Pred.	Dev.
1	11.63	13.35	1.72	13.34	1.71	13.20	1.57	19.78	8.15
2	16.92	15.73	1.19	14.88	2.04	15.33	1.59	22.30	5.38
3	21.36	21.29	0.07	22.45	1.09	22.09	0.73	25.50	4.14
4	28.40	25.88	2.52	25.02	3.38	24.83	3.58	29.13	0.73
5	33.21	33.41	0.20	34.66	1.45	34.53	1.32	34.37	1.16
6	39.46	38.60	0.86	40.10	0.64	40.04	0.58	39.38	0.08
7	42.43	45.43	3.00	46.24	3.81	46.09	3.66	44.98	2.55
8	48.33	48.76	0.43	48.11	0.22	48.44	0.11	49.23	0.90
9	56.45	55.01	1.44	51.71	4.74	51.75	4.70	53.94	2.51
10	60.41	62.90	2.49	60.85	0.44	61.24	0.83	59.31	1.10
11	65.91	66.99	1.08	68.41	2.50	69.32	3.41	64.76	1.15
12	73.41	72.70	0.71	72.22	1.19	72.52	0.89	70.17	3.24
13	77.34	79.75	2.41	79.65	2.31	79.69	2.35	75.50	1.84
14	85.42	83.45	1.97	83.78	1.64	83.60	1.82	80.65	4.77
15	89.42	91.03	1.61	88.61	0.81	87.91	1.51	86.75	2.67
16	92.50	94.41	1.91	94.74	2.24	94.28	1.78	91.63	0.87
17	100.00	97.26	2.74	98.63	1.37	97.48	2.52	96.21	3.79
18	11.34	11.04	0.30	9.55	1.79	9.35	1.99	18.58	7.24
19	16.32	15.23	1.09	15.20	1.12	15.10	1.22	21.08	4.76
20	21.93	20.38	1.55	19.85	2.08	19.79	2.14	24.52	2.59
21	24.20	26.40	2.20	25.32	1.12	25.04	0.84	29.06	4.86
22	31.21	29.26	1.95	30.15	1.06	30.34	0.87	32.93	1.72
23	37.98	36.76	1.22	33.90	4.08	33.97	4.01	37.63	0.35
24	47.98	43.53	4.45	41.09	6.89	41.13	6.85	42.66	5.32
25	54.45	53.85	0.60	59.05	4.60	60.44	5.99	50.11	4.34
26	60.92	60.57	0.35	55.59	5.33	55.06	5.86	57.23	3.69
27	68.29	67.48	0.81	67.34	0.95	68.15	0.14	64.25	4.04
28	76.53	74.73	1.80	73.72	2.81	73.13	3.40	70.55	5.98
29	80.45	82.56	2.11	81.60	1.15	81.97	1.52	77.41	3.04
30	83.57	86.29	2.72	85.99	2.42	85.87	2.30	83.32	0.25
31	87.54	89.39	1.85	87.72	0.18	87.32	0.22	87.94	0.40
32	90.52	92.98	2.46	96.23	5.71	97.14	6.62	91.35	0.83
33	96.44	95.54	0.90	97.94	1.50	97.36	0.92	94.44	2.00
34	100.00	100.72	0.72	99.55	0.45	99.09	0.91	98.50	1.50
35	8.65	7.21	1.44	8.33	0.32	8.10	0.55	15.83	7.18
36	11.71	12.28	0.57	11.25	0.46	11.01	0.70	18.52	6.81
37	13.40	15.39	1.99	15.81	2.41	15.82	2.42	21.16	7.76
38	15.03	17.71	2.68	16.40	1.37	16.79	1.76	24.01	8.98

Continued Table 3

Case	Actual output	LS-SVM _{AT}		LS-SVM _L		LS-SVM _Q		ESIM	
		Pred.	Dev.	Pred.	Dev.	Pred.	Dev.	Pred.	Dev.
39	21.98	19.72	2.26	18.50	3.48	17.94	4.04	25.95	3.97
40	24.24	26.72	2.48	27.20	2.96	27.85	3.61	29.30	5.06
41	30.41	29.13	1.28	29.52	0.89	30.03	0.38	32.59	2.18
42	35.81	35.99	0.18	33.49	2.32	33.69	2.12	37.38	1.57
43	42.49	41.45	1.04	41.89	0.60	42.11	0.38	41.70	0.79
44	52.94	48.54	4.40	50.00	2.94	50.64	2.30	47.36	5.58
45	60.52	59.01	1.51	59.93	0.59	60.69	0.17	54.46	6.06
46	69.21	66.63	2.58	58.89	10.32	54.87	14.34	62.07	7.14
47	74.37	75.47	1.10	75.14	0.77	72.24	2.13	70.31	4.06
48	76.27	80.52	4.25	78.42	2.15	78.44	2.17	76.83	0.56
49	87.50	82.65	4.85	84.09	3.41	85.09	2.41	81.53	5.97
50	90.53	92.99	2.46	86.05	4.48	84.40	6.13	87.42	3.11
51	100.00	95.23	4.77	96.78	3.22	96.15	3.85	92.31	7.69

Note: Pred. denotes predicted output;
Dev. denotes result deviation.

series data. LS-SVM_{AT} makes prediction of future project cash demands by case learning of patterns in the past. Moreover, DE searching algorithm is utilized in the cross validation process to identify the most appropriate tuning parameters without the need of trial-and-error process. Experimental result and performance comparison have proved the strong potential of the new inference model.

Currently, LS-SVM_{AT} has a limitation: the model is built using a single database collected from one construction contractor in Taipei. Although the data are quiet homogeneous and capable of facilitating cash flow estimation effectively, more historical cases from different contractors should be incorporated to enhance the generalization of the prediction model. On the other hand, all of the recorded data used for cash flow prediction are high-rise building projects. Hence, data collected from other types of construction project, such as highway and tunnelling structure, can be worth investigated. It is because other project types may possess different characteristics. Nevertheless, the procedure of collecting new data cases requires time and effort. Therefore, we would like to consider these to be promising future research directions.

References

- An, S.-H.; Park, U. Y.; Kang, K.-I.; Cho, M.-Y.; Cho, H.-H. 2007. Application of support vector machines in assessing conceptual cost estimates, *Journal of Computing in Civil Engineering* ASCE 21(4): 259–264. [http://dx.doi.org/10.1061/\(ASCE\)0887-3801\(2007\)21:4\(259\)](http://dx.doi.org/10.1061/(ASCE)0887-3801(2007)21:4(259))
- Becerra, R. L.; Coello, C. A. C. 2006. Cultured Differential Evolution for constrained optimization, *Computer Methods in Applied Mechanics and Engineering* 195(33–26): 4303–4322. <http://dx.doi.org/10.1016/j.cma.2005.09.006>
- Bishop, C. 2006. *Pattern recognition and machine learning*. Springer Science+Business Media. 738 p.
- Boussabaine, A. H.; Kaka, A. P. 1998. A neural networks approach for cost flow forecasting, *Construction Management and Economics* 16(4): 471–479. <http://dx.doi.org/10.1080/014461998372240>
- Cheng, M.-Y.; Peng, H.-S.; Wu, Y.-W.; Chen, T.-L. 2010. Estimate at completion for construction projects using evolutionary support vector machine inference model, *Automation in Construction* 19(5): 619–629. <http://dx.doi.org/10.1016/j.autcon.2010.02.008>
- Cheng, M.-Y.; Wu, Y.-W. 2009. Evolutionary support vector machine inference system for construction management, *Automation in Construction* 18(5): 597–604. <http://dx.doi.org/10.1016/j.autcon.2008.12.002>
- De Brabanter, K.; Karsmakers, P.; Ojeda, F.; Alzate, C.; De Brabanter, J.; Pelckmans, K.; De Moor, B.; Vandewalle, J.; Suykens, J. A. K. 2010. *LS-SVMlab toolbox user's guide version 1.8*. Internal Report 10-146. Leuven: ESAT-SISTA, K.U.Leuven. 115 p.
- Guo, Z.; Bai, G. 2009. Application of least squares support vector machine for regression to reliability analysis, *Chinese Journal of Aeronautics* 22(2): 160–166. [http://dx.doi.org/10.1016/S1000-9361\(08\)60082-5](http://dx.doi.org/10.1016/S1000-9361(08)60082-5)
- Hwee, N. G.; Tiong, R. L. K. 2002. Model on cash flow forecasting and risk analysis for contracting firms, *International Journal of Project Management* 20(5): 351–363. [http://dx.doi.org/10.1016/S0263-7863\(01\)00037-0](http://dx.doi.org/10.1016/S0263-7863(01)00037-0)
- Jiang, A.; Issa, R. R. A.; Malek, M. 2011. Construction project cash flow planning using the pareto optimality efficiency network model, *Journal of Civil Engineering and Management* 17(4): 510–519. <http://dx.doi.org/10.3846/13923730.2011.604537>
- Khosrowshahi, F.; Kaka, A. P. 2007. A decision support model for construction cash flow management, *Computer-Aided Civil and Infrastructure Engineering* 22(7): 527–539. <http://dx.doi.org/10.1111/j.1467-8667.2007.00508.x>
- Lin, C.-F.; Wang, S.-D. 2002. Fuzzy support vector machines, *IEEE Transactions on Neural Networks* 13(2): 464–471. <http://dx.doi.org/10.1109/72.991432>

- Mezura-Montes, E.; Coello, C. A. C.; Tun-Morales, E. I.; Computación, S. D.; Tabasco, V. 2004. Simple feasibility rules and Differential Evolution for constrained optimization, in *Proc. of the 3rd Mexican International Conference on Artificial Intelligence (MICAI 2004)*, 26–30 April 2004, Mexico City, Mexico, 707–716.
- Nam, H.; Han, S. H.; Kim, H. 2007. Time series analysis of construction cost index using wavelet transformation and a neural network, in *Proc. of the 24th International Symposium on Automation & Robotics in Construction (ISARC 2007)*, 19–21 September 2007, Madras, India, 453–456.
- Price, K. V.; Storn, R. M.; Lampinen, J. A. 2005. *Differential Evolution a practical approach to global optimization*. Springer-Verlag. 538 p.
- Russell, J.; Jaselskis, E.; Lawrence 1997. Continuous assessment of project performance, *Journal of Construction Engineering and Management* ASCE 123(1): 64–71. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1997\)123:1\(64\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1997)123:1(64))
- Samarasinghe, S. 2006. *Neural networks for applied sciences and engineering*. Taylor and Francis. 570 p.
- Samui, P.; Lansivaara, T.; Kim, D. 2011. Utilization relevance vector machine for slope reliability analysis, *Applied Soft Computing* 11(5): 4036–4040. <http://dx.doi.org/10.1016/j.asoc.2011.03.009>
- Sapankevych, N.; Sankar, R. 2009. Time series prediction using support vector machines: a survey, *IEEE Computational Intelligence Magazine* 4(2): 24–38. <http://dx.doi.org/10.1109/MCI.2009.932254>
- Storn, R.; Price, K. 1997. Differential Evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11(1997): 341–359. <http://dx.doi.org/10.1023/A:1008202821328>
- Suykens, J. A. K.; Gestel, J. V.; Brabanter, J. D.; Moor, B. D.; Vandewalle, J. 2002. *Least squares support vector machines*. Leuven: World Scientific Publishing Co. Pte. Ltd. 84 p.
- Suykens, J. A. K.; Vandewalle, J. 1999. Least squares support vector machine classifiers, *Neural Processing Letters* 9(3): 293–300. <http://dx.doi.org/10.1023/A:1018628609742>
- Williams, T. P. 1994. Predicting changes in construction cost indexes using neural networks, *Journal of Construction Engineering and Management* ASCE 120(2): 306–320. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1994\)120:2\(306\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1994)120:2(306))
- Yu, L.; Chen, H.; Wang, S.; Lai, K. K. 2009. Evolving least squares support vector machines for stock market trend mining, *IEEE Transactions on Evolutionary Computation* 13(1): 87–102. <http://dx.doi.org/10.1109/TEVC.2008.928176>

Min-Yuan CHENG. He is currently a Professor at the Department of Civil and Construction Engineering, National Taiwan University of Science and Technology. He holds lectures in Construction Automation and Construction Process Reengineering. He has published many papers in various international journals such as *Journal of Civil Engineering and Management*, *Automation in Construction*, *Journal of Construction Engineering and Management*, and *Expert Systems with Applications*. His research interests include management information system, applications of artificial intelligence, and construction management process reengineering.

Nhat-Duc HOANG. He is currently a researcher and lecturer at the Institute of Research and Development and Faculty of Civil Engineering, Duy Tan University. He got the MSc and PhD degrees at National Taiwan University of Science and Technology. He has published papers in various international journals including *Journal of Civil Engineering and Management*, *Automation in Construction*, *Journal of Computing in Civil Engineering*, *Knowledge-Based Systems*, *Journal of Construction Engineering*, and *Tunnelling and Underground Space Technology*. His research focuses on applications of Artificial Intelligence in construction engineering and management.

Yu-Wei WU. He is currently a Post-doctoral Researcher at the Department of Civil and Construction Engineering, National Taiwan University of Science and Technology. His research papers appear in many international journals such as *Automation in Construction*, *Expert Systems with Applications*, and *Engineering Applications of Artificial Intelligence*. His research interests include construction management and information technology.