Prediction of project cash flow using time-depended evolutionary LS-SVM inference model

Min-Yuan Cheng, Nhat-Duc Hoang *, and Yu-Wei Wu

Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
** Corresponding author (ducxd85@yahoo.com)*

Purpose The ability to predict cash demand is crucial for the operation of construction companies. Reliable cash flow prediction during the execution phase can help managers to avoid cash shortages and to control project cash flow effectively. **Method** This paper presents a new inference model, CF-ELSIMT, for cash flow forecasting. The developed CF-ELSIMT utilizes weighted Least Squares Support Vector Machine (wLSSVM) as a supervised learning technique to generalize the mapping function between input and output of cash flow time series. A novel dynamic time function (TF) is employed to determine the weighting values associated with data in different time periods. The dynamic TF allows the model to deal with distinct characteristics in cash flow time series. To optimize the model's tuning parameters, the new inference model incorporates Differential Evolution (DE) as the search engine. In addition, a machine-learning-based interval estimation (MLIE) approach is used to arrive at the prediction interval of forecasted cash demand. **Results & Discussion** The CF-ELSIMT provides construction planners with a point estimate coupled with the lower and upper prediction intervals. Experimental results and comparisons have demonstrated that the newly established model has enhanced the forecasting accuracy.

***Key words**: construction management, weighted LS-SVM, cash flow forecasting, cost control*

INTRODUCTION

In construction industry, cash is a critical factor that imposes influence on project profitability[1]. Poor cash flow control can lead to project failure for contractors due to liquidity shortage for supporting their daily activities[2]. Hence, reliable prediction of cash flow time series over the course of a construction project is beneficial since it puts the project manager in a better position to identify potential problems and to develop appropriate strategies to mitigate the negative effects of such on overall project success.

Due to the importance of the problem at hand, various models have been proposed to predict the project cash flow. Boussabaine and Kaka employed neural networks in cash flow forecasting and control[3]. In addition, fuzzy logic based techniques have also been applied to increase the effectiveness of cash flow analysis conducted under uncertain conditions[4, 5]. Park et al. proposed a forecasting model for construction projects that considered both variable cost weights and time lag[6]. However, most of previous models were developed to assist manager in the pre-tendering or planning stage of a project, few researches have addressed the dynamic and time-depended nature of the cash flow prediction problem.

Additionally, prediction of cash flow is often stated in the form of a point forecast[7, 8]. However, in practice, project managers require not only accurate forecasts of cash flow but also the uncertainty associated with the predictions. Interval estimation includes the upper and lower limits between which a predicted variable is expected to lie with a certain level of confidence. The range restricted by those limits is known as prediction interval (PI) (see Fig. 1). Thus, incorporating prediction uncertainty expressed by prediction interval can help improve the reliability and the credibility of the model outputs.[9]

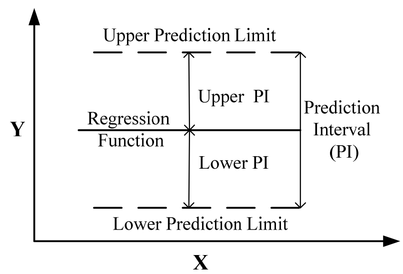

Fig. 1 Prediction Interval

Recently, a new framework for achieving prediction interval (PI) which is based on machine learning technique has been established by Thresha and Solomatine[9]. The proposed machine learning based interval estimation (MLIE) does not require any assumption and prior knowledge of input data or model error distribution. In their research work [9], the superiority of the MLIE over existing methods is exhibited. Thus, it is beneficial to incorporate this approach into a forecasting model to obtain the interval estimation.

Proposed by Suykens et al.[10, 11], WLS-SVM is an advanced machine learning technique which possesses many advanced features. In WLS-SVM's training process, a least squares cost function is proposed to obtain a linear set of equations in the dual space. Consequently, it is required to deal with a set of linear system which can be efficiently solved by iterative methods such as conjugate gradient[12]. Furthermore, in this approach, a weighting value is assigned to each error variable[13]. This feature allows each training data point to contribute differently to the establishment of the regression function and facilitate WLS-SVM to better deal with time series problems such as cash flow prediction.

Another issue in the field of AI is the mechanism for setting models' control parameters. In practice, identifying model's parameters often requires time-consuming trial-and-error processes. Thus, hybridizing the machine learning techniques with an evolutionary algorithm (EA) is a prevalent research direction[14]. Among EA techniques, Differential Evolution (DE) [15] is a population-based stochastic search engine, which is efficient and effective for global optimization in the continuous domain. Superior performance of DE over other algorithms has been verified in many reported research works. [15, 16]

Therefore, purpose of this study is to hybridize WLS-SVM, MLIE, APLF, and DE to establish a new inference model for predicting time-cost curve of construction projects. Since the cash flow data are time-dependent, the integrated model employs WLS-SVM to infer the mapping between past and future instances of the time-cost curve. Moreover, APLF is used to determine the weighting values associated with each data. In order to automatically identify the tuning parameters, the new inference model utilizes DE. Additionally, MLIE approach is deployed to calculate prediction intervals of forecasted outputs.

The second section of this paper reviews related literature on WLS-SVM, MLIE, and DE. In the third section, detail of the proposed adaptive time function is introduced. The framework of the proposed model CF-ELSIM$_T$ is depicted in the forth section. The fifth section demonstrates the experimental results. Conclusion on our study is mentioned in the final section.

LITERATURE REVIEW

Weighted Least Squares Support Vector Machine (WLS-SVM)

This section reviews the formulation of WLS-LSVM, proposed by Suykens et al.[13]. Consider the following model, which describes the mapping relationship between a response variable and independent variables:

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

where $x \in R^n$, $y \in R$, and $\phi(x): R^n \to R^{nh}$ is the mapping to the high dimensional feature space. The formulation of WLS-SVM, given a training dataset $\{x_k, y_k\}_{k=1}^N$, can be given as follows: [13, 17]

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

Subjected to $y_k = w^T \phi(x_k) + b + e_k$, $k = 1,\dots,N$

where $e_k \in R$ are error variables; $\gamma > 0$ denotes a regularization constant; $s_k \in [0,1]$ is a weighting value associated with an error variable.

The above optimization problem stated in (2) can be solved by constructing the Lagrangian and deriving the following dual problem. [10]

The Lagrangian is given by:

$$L(w,b,e;\alpha) = J_p(w,e) - \sum_{k=1}^{N} \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\} \qquad (3)$$

where α_k are Lagrange multipliers. The conditions for optimality are given by:

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \to w = \sum_{k=1}^{N} \alpha_k \phi(x_k) \\ \dfrac{\partial L}{\partial b} = 0 \to \sum_{k=1}^{N} \alpha_k = 0 \\ \dfrac{\partial L}{\partial e_k} = 0 \to \alpha_k = \gamma s_i e_k, \; k = 1,\dots,N \\ \dfrac{\partial L}{\partial \alpha_k} = 0 \to w^T \phi(x_k) + b + e_k - y_k = 0, \; k = 1,\dots N \end{cases} \qquad (4)$$

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

$$\begin{bmatrix} 0 & -Y^T \\ Y & \omega \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad (5)$$

where

$$Y = (y_1, y_2, ..., y_N)^T \quad (6)$$

$$\alpha = (\alpha_1; ...; \alpha_N) \quad (7)$$

$$\omega_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) + (s_i \gamma)^{-1} I \quad (8)$$

$$1 = (1,1,...,1)^T \quad (9)$$

And the kernel function is applied as follow:

$$K(x_k, x_l) = \phi(x_k)^T \phi(x_l) \quad (10)$$

The resulting LS-SVM model for function estimation is expressed as:

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x_k, x_l) + b \quad (11)$$

where α_k and *b* are the solution to the linear system (5). The kernel function that is often utilized is Radial Basis Function (RBF) kernel. Description of RBF kernel is given as follow:

$$K(x_k, x_l) = \exp\left(\frac{\|x_k - x_l\|^2}{2\sigma^2}\right) \quad (12)$$

where σ denotes the kernel function parameter.

Machine-learning Based Interval Estimation

This section reviews the machine learning based interval estimation (MLIE), which was proposed by Thresha and Solomatine.[9] The MLIE approach[9] is described in Fig. 2. At first, the point estimation process is carried out. A regression model is implemented to infer the mapping function between input data and the corresponding outputs. The input data points are then separated into different clusters that have similar historical residuals, which are obtained from point estimation process, using fuzzy c-means clustering (FCMC) [18]. When applying FCMC, the number of clusters is commonly selected so that it results in a minimum value of Xie-Beni index. [19]

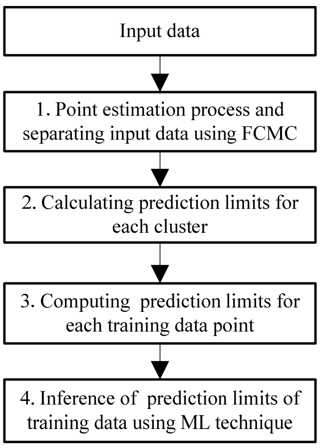

Fig. 2 Machine learning based interval estimation (MLIE)

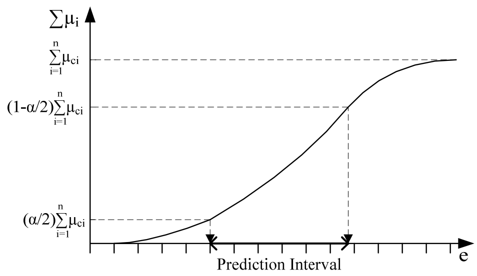

Fig. 3 Calculating Prediction Interval for each cluster

In the second step, the lower and upper prediction intervals (PIs) for each cluster are computed. Given a certain level of confidence (e.g. 95% or α is 5%), the PIs for each cluster are calculated from empirical distributions of the corresponding historical residuals (*e*). To construct $(100-\alpha)\%$ prediction interval (PI), the $(\alpha/2)\times 100$ and $(1-(\alpha/2))\times 100$ percentile values are taken from empirical distribution of residuals for lower and upper prediction intervals, respectively (see Figure 3).The mathematical expression for calculating lower and upper PIs for cluster *i* (PI_{ci}^L and PI_{ci}^U) is given as follows:

$$PI_{ci}^L = e_j \qquad j : \sum_{k=1}^{j} \mu_{i,k} < \frac{\alpha}{2} \sum_{j=1}^{n} \mu_{i,j} \quad (13)$$

$$PI_{ci}^U = e_j \qquad j : \sum_{k=1}^{j} \mu_{i,k} > (1-\frac{\alpha}{2}) \sum_{j=1}^{n} \mu_{i,j} \quad (14)$$

where *j* is the index of the sorted data point that satisfies the corresponding inequalities. e_j denotes historical residuals of sorted data point *j*. And $\mu_{i,j}$ is membership grade of data point *j* to cluster *i*.

The third step is to calculate the PI for each training data point using the weighted mean of PIs of each cluster:

$$PI_j^L = \sum_{i=1}^{c} \mu_{i,j} \times PI_{ci}^L \quad (15)$$

$$PI_j^U = \sum_{i=1}^{c} \mu_{i,j} \times PI_{ci}^U \quad (16)$$

where PI_j^L and PI_j^U are lower and upper PIs for data point *j*.

Prediction limits (PLs) for each data point are computed as follows:

$$PL_j^L = y_i + PI_j^L \quad (17)$$

$$PL_j^U = y_i + PI_j^U \quad (18)$$

where PL_j^L and PL_j^U are lower and upper PLs of predicted output *j*.
In the final step, a machine learning (ML) technique (e.g. LS-SVM) can be deployed to learn the mapping functions between the input data and the computed PLs for training data. PLs for testing data can be inferred using those underlying functions.

Differential Evolution

Differential evolution (DE) is an Evolutionary Algorithm which is designed for real parameter optimization.[15] DE algorithm is based 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).[20] 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.[21] The algorithm of differential evolution is shown in Fig. 4. 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*.

Initialize population of NP individuals
Do
For each individual *j* in the population
Generate three random integers r_1, r_2, and $r_3 \in (1, NP)$
with $r_1 \neq r_2 \neq r_3 \neq j$
Generate random integer $i_{rand} \in (1, D)$
For each parameter *i*

$$U_{j,i,g} = \begin{cases} X_{j,r3,g} + F \times (X_{j,r1,g} - X_{j,r2,g}) & \text{if } rand_j(0,1) < Cr \text{ or } j = rnb(i) \\ X_{j,i,g} & \text{otherwise} \end{cases}$$

End For
Replace X_i with the offspring U_i if U_i is better
End For
Until the stopping condition is met

Fig. 4 Differential Evolution optimization algorithm

In the selection process, the trial vector is compared to the target vector (or the parent).[16] 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 follow:

$$X_{i,g+1} = \begin{cases} U_{i,g} \ \ if \ \ f(U_{i,g}) \le f(X_{i,g}) \\ X_{i,g} \ \ if \ \ f(U_{i,g}) > f(X_{i,g}) \end{cases} \tag{19}$$

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.

ADAPTIVE PIECEWISE LINEAR FUNCTION FOR WEIGHTING TIME SERIES DATA

Real-world time series data are often unbalanced due to the fact that recent data can provide more relevant information than distance ones. Therefore, time series data should to be weighted differently. Instead of using fixed time functions, this study proposes an adaptive piecewise linear function (APLF) for weighting data.
The role of the APLF is to determine a weighting value to each data point in the training process. The time function assigns small weighting values for data points at the initial phase of a project. Meanwhile, data points recorded at the later phase are coupled with greater weighting values (see Fig. 5). Using the proposed APLF, the time horizon of a completed project is divided into several domains. Each domain is characterized by a linear time function described as follow:

$$s_i^k = s_o + a_k(t_i), k = 1, \forall i \in R_1 \tag{20}$$

$$s_i^k = \max_{\forall i \in R_{k-1}}(s_i^{k-1}) + a_k(t_i), k \ge 1, \ \forall i \in R_k \tag{21}$$

$$0 \le a_k \le a_k^{\max} \quad \text{where:} \tag{22}$$

$$a_k^{\max} = \frac{(1 - s_o)}{n_k}, \ k = 1 \tag{23}$$

$$a_k^{\max} = \frac{(1 - \max_{\forall i \in R_{k-1}}(s_i^{k-1}))}{n_k}, \ k \ge 1 \tag{24}$$

where s_i^k denote the weight value for data point *i* in the k^{th} domain. s_o, varying between 0 and 1, is the initial value of the time function in the first domain. a_k represents the slope value of the time function in the k^{th} domain. R_k is the set of time periods in the k^{th} domain. And, n_k is the index of the last time period in the k^{th} domain. For instance, if a domain *j* contains four time periods: 3, 4, 5, and 6, the corresponding n_j is 6. The Eq. (20) and (21) calculate the weighting value for each time period. The Eq. (22), (23), and (24) control the magnitude of the slope parameters so that every weighting value is of the range [0, 1].

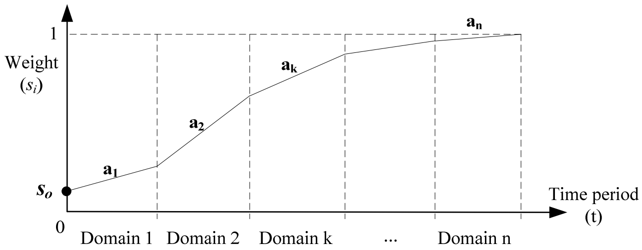

Fig. 5 APLF for weighting time series data

For the first domain, the time function has two free parameters: the initial value (s_o) and the slope (a_1). The time function for other domain only needs the slope parameter to specify its shape. Consider the case in which each project has *M* completion periods, the project duration is separated into *n* domains. Hence, there are *n+1* tuning parameters that needed to be specified. When the APLF is integrated into the overall model, its tuning parameters are automatically optimized by the search engine.

CASH FLOW PREDICTION USING TIME-DEPENDED EVOLUTIONARY LS-SVM INFERENCE MODEL (CF-ELSIM$_T$)

This section dedicates in describing the proposed prediction model, named as CF-ELSIM$_T$, in detail. The establishment of the model (see Fig. 6) is accomplished by a fusion of various prevalent AI techniques. CF-ELSIM$_T$ employs WLS-SVM as the supervised learning algorithm for mining the implicit patterns in the series. Moreover, the new forecasting model incorporates the MLIE for achieving interval prediction. Finally, DE, an evolutionary optimization algorithm, is utilized to automatically identify the optimal values of tuning parameters.

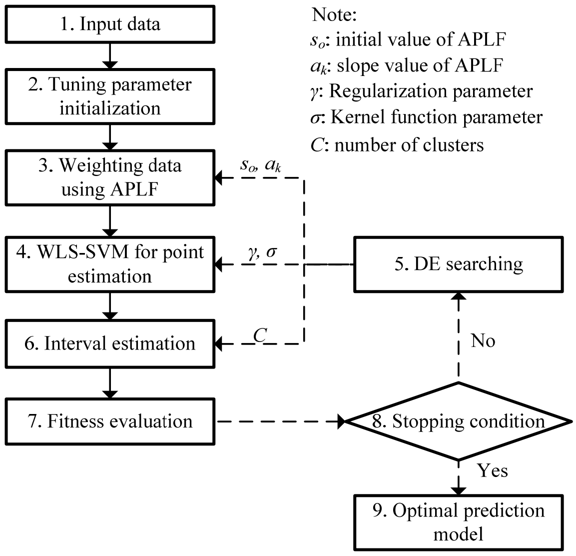

Fig. 6 Cash Flow Forecasting Using Time-depended Evolutionary Least Squares Support Vector Machine Inference Model (CF-ELSIM$_T$)

The database used in the paper, collected from a construction contractor in Taipei, was generated in the process of executing high rise projects between 1996 and 2006. This database contains percentage of expenditure cash flow taken from 13 completed construction projects reported in. [22] Table 1 illustrates the cash flow data for one project in the database. The CF-ELSIM$_T$ utilizes 8 projects as training set, 2 projects as validating set, and 3 projects as testing set. Standard cumulative cost-time curves were employed to model cash flow prediction.

(1) *Input data*: The model CF-ELSIM$_T$ takes the cash flow time series as its input. There are 17 cases inherent 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.

Table 1. Example of expenditure cash flow for one construction project

Case	Input pattern			Output
	1	2	3	
	1st period	2nd period	3rd period	4th period
1	0.2	2.7	5.1	9.7
2	2.7	5.1	9.7	12.2
3	5.1	9.7	12.2	15.3
…	…	…	…	…
16	75.5	83.5	87.6	92.3
17	83.5	87.6	92.3	100.0

(2) *Tuning parameter initialization*: The construction of the prediction model is dependent on a set of tuning parameters (see Table 2). The parameters of APLF consist of the initial value (s_o) and the slope value (a_k), which are needed for weighting data. The regularization parameter (γ) and the kernel function parameter (σ) are required for the WLS-SVM. The number of clusters (C) is needed to be specified for the fuzzy c-means clustering process.

Table 2. Ranges of model's tuning parameters

Tuning parameter		**Lower bound**	**Upper bound**
Initial value of APLF	s_o	0	1
Slope value of APLF	a_k	0	1
Regularization parameter	γ	0.001	10000
Kernel function parameter	σ	0.001	1000
Number of cluster	C	2	10

(3) *Adaptive piecewise linear function (APLF) for weighting data*: Each training data point is weighted according to the APLF. It is noted that the weights computed from the APLF ranges from a relatively small starting value s_o to 1. Hence, the most recent data point is treated as the most important and thus, received the highest value of 1. Meanwhile, the most distant data point is considered as the least important and given the smallest value of s_o.

(4) *WLS-SVM for point estimation*: In this step, LS-SVM is deployed to learn the mapping function between the input (X) and the output (Y) derived at the previous step. The training process requires the two parameters γ and σ that are acquired from the DE searching. These parameters play an important role in determining the model's prediction accuracy.

(5) *DE searching*: At each generation, the optimizer carries out the mutation, crossover, and selection processes to guide the population to the optimal solution.

(6) *Interval estimation*: This step employs the MLIE approach established by Thresha and Solomatine [9]. In the fuzzy clustering process, the search engine is employed to find the number of cluster (C). After the prediction limits for each training data point are computed, two LS-SVM models are employed to learn the regression function between input data and the two PLs. The tuning parameters of LS-SVM for interval estimation are identical to that of LS-SVM for point estimation, which are automatically identified by the search engine.

(7) *Fitness evaluation*: In ELSIM, in order to determine the optimal set of tuning parameters, the following objective function is used in the step of fitness function evaluation:

$$F_{fitness} = \alpha \times E_{tr} + \beta \times E_{va} + \theta \times S \qquad (25)$$

In Eq. (16), α, β, and θ are weighting coefficients. E_{tr} and E_{va} denotes the training error and validating error, respectively. The training and validating errors herein are Root Mean Squared Error (RMSE). S represents the Xie-Beni index [19], which is calculated as followed:

$$S = \frac{\sum_{i=1}^{C}\sum_{j=1}^{n} \mu_{i,j}^{2} \left\| V_i - X_j \right\|^2}{n \times \min_{i,j} \left\| V_i - V_j \right\|^2} \qquad (26)$$

where X_j denotes the data point *j*. V_i is the center of cluster *i*. And, *n* is the number of data points. (8) *Stopping condition*: The DE's optimization process terminates when the maximum number of generation is achieved.

(9) *Optimal prediction model*: When the program terminates, the optimal set of tuning parameters has been successfully identified. The CF-ELSIM$_T$ is ready to carry out forecasting tasks.

EXPERIMENTAL RESULT

This section validates the performance of the proposed prediction model. To illustrate that CF-ELSIM$_T$ is capable of delivering accurate and reliable results, the outcome of the proposed model is benchmarked with Evolutionary Support Vector Machine Inference Model (ESIM).[14] In order to evaluate the accuracy of EAC point estimation, RMSE is employed (see Table 3). RMSE of CF-ELSIM$_T$ for training is 0.013. Moreover, it is noticed that CF-ELSIM$_T$ utilized the APLF for weighting data; the optimal shape of the weighting function is shown in Fig. 7. Meanwhile, RMSE for testing projects 1, 2, and 3 are 0.020, 0.041 and 0.024, respectively. It is observable that the new model outperformed the benchmark approach in prediction accuracy since the prediction error of ESIM for two testing projects are 0.036, 0.048, and 0.052.

Table 3. Result comparison

Model	Training RMSE	Testing project 1 RMSE	Testing project 2 RMSE	Testing project 3 RMSE
ESIM	0.045	0.036	0.048	0.052
CF-ELSIM$_T$	0.013	0.020	0.041	0.024

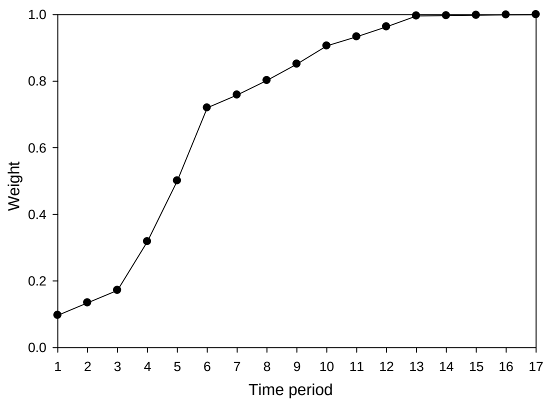

Fig. 7 Optimal APLF

Table 4. Results of interval estimation using CF-ELSIM$_T$

Interval prediction using CF-ELSIM$_T$	LOC (%)	PICP (%)	MPI
	90	90	0.09
	95	96	0.13

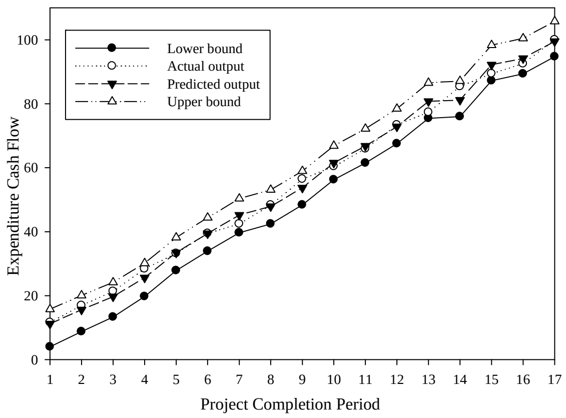

Fig. 8 Prediction result of testing project 1 using CF-ELSIM$_T$ (95% LOC)

Furthermore, to assess performance of the constructed prediction interval, Prediction Interval Coverage Probability (PICP) and Mean Width of Prediction Interval (MPI) are utilized (see Table 4). When the level of confidence (LOC) is 90%, the PICP and the MPI of the proposed model for the two testing projects are 90% and 0.09, respectively. Those two values are calculated to be 96% and 0.13 in the case of 95% LOC. Fig. 8 illustrate the result of interval prediction of CF-ELSIM$_T$ for one testing project with 95% LOC. Observably, the proposed model is accurate in interval forecast of project cash flow. It has achieved acceptable values of PICP value corresponding to relatively small values of MPI.

CONCLUSION

This paper has presented a new prediction model, named as CF-ELSIM$_T$, to assist construction managers in dealing with forecasting of project cash flow. The proposed model was developed by a fusion of various advanced AI techniques, namely: WLS-SVM, APLF, MLIE, and DE. The WLS-SVM is utilized to infer the input/output mapping function of cash flow data. The APLF helps the model to be more appropriate in coping with real-world time-dependent data. Meanwhile, to address the uncertainty of prediction results, the model integrates the MLIE approach. Using MLIE, the prediction interval is constructed by evaluating the uncertainty inherent in the data set, without any assumption or prior knowledge of model's error. Moreover, DE searching algorithm is utilized to identify the most appropriate set of tuning parameters without the need of experience or trial-and-error process in parameter setting.

Consequently, the model's output consists of the point estimation coupled with the lower and upper prediction intervals, given a certain level of confidence, to emphasize the forecasting uncertainty. Furthermore, the newly developed model has the ability to operate automatically without human intervention and domain knowledge. Simulation result and performance comparison have proved the strong potential of CF-ELSIM$_T$ as an alternative for cash flow forecasting.

Currently, CF-ELSIM$_T$ has a limitation is that the model is built using the 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 recorded projects are high-rise buildings; hence, construction projects of other types, such as highway or steel structures can be worth investigated. It is because other project types may possess different characteristics. Nevertheless, the processes of collecting new data cases are of great effort and time-consuming. Hence, we would like to consider these to be promising future research directions.

References

1. Hwee, N.G., Tiong, R.L.K., "Model on cash flow forecasting and risk analysis for contracting firms", *International Journal of Project Management*, vol. 20(5), pp. 351-363, 2002.
2. Khosrowshahi, F., Kaka, A.P., "A Decision Support Model for Construction Cash Flow

Management", *Computer-Aided Civil and Infrastructure Engineering*, vol. 22(7), pp. 527-539, 2007.

3. Boussabaine, A.H., Kaka, A.P., "A neural networks approach for cost flow forecasting", *Construction Management and Economics,* vol. 16(4): pp. 471-479, 1998.
4. Maravas, A., Pantouvakis, J.P, "Project cash flow analysis in the presence of uncertainty in activity duration and cost" *International Journal of Project Management,* vol. 30(3), pp. 374-384.
5. Boussabaine, A.H., Elhag, T., "Applying fuzzy techniques to cash flow analysis", *Construction Management and Economics,* vol. 17(6), pp. 745-755, 1999.
6. Park, H.K., Han, S.H. ,Russell, J.S., "Cash Flow Forecasting Model for General Contractors Using Moving Weights of Cost Categories", *Journal of Management in Engineering,* vol. 21(4), pp. 164-172, 2005.
7. Cheng, M.-Y., Roy, A.F.V., "Evolutionary fuzzy decision model for cash flow prediction using time-dependent support vector machines", *Int. J. Proj. Manage.,* vol. 29(1), pp. 56-65, 2011.
8. Cheng, M.-Y., Tsai, H.-C., Sudjono, E., "Evolutionary fuzzy hybrid neural network for project cash flow control", *Eng. Appl. Artif. Intel.,* vol. 23(4), pp. 604-613, 2010.
9. Shrestha, D.L., Solomatine, D.P., "Machine learning approaches for estimation of prediction interval for the model output", *Neural Networks,* vol. 19(2), pp. 225-235, 2006.
10. Suykens, J., et al., "Least Square Support Vector Machines", *World Scientific Publishing Co. Pte. Ltd.,* 2002.
11. Brabanter, de, K., et al., "LS-SVMlab Toolbox User's Guide version 1.8.", *Internal Report 10-146,* ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2010.
12. Wang, H., Hu, D., "Comparison of SVM and LS-SVM for Regression", *Neural Networks and Brain,* ICNN&B '05, International Conference, 2005.
13. Suykens, J.A.K., et al., "Weighted least squares support vector machines: robustness and sparse approximation", *Neurocomputing*, vol. 48, pp. 85-105, 2002.
14. Cheng, M.-Y., Wu, Y.-W., "Evolutionary support vector machine inference system for construction management", *Autom.Constr.,* vol. 18(5), pp. 597-604, 2009.
15. Price, K.V., Storn, R.M., Lampinen, J.A., "Differential Evolution A Practical Approach to Global Optimization", *Springer-Verlag*, 2005.
16. Storn, R., Price, K., "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", *J. Global Optim.* , vol. 11, pp. 341–359, 1997.
17. Lean, Y., Keune, L. Kin, Wane, S., "Credit Risk Assessment with Least Squares Fuzzy Support Vector Machines", Data Mining Workshops, ICDM Workshops 2006, *Sixth IEEE International Conference,* 2006.
18. Bezdek, J.C., "Pattern recognition with fuzzy objective function algorithms", *Norwell,* MA: Kluwer, 1981.
19. Xie, X.L., Beni, G., "A validity measure for fuzzy clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1991.
20. Becerra, R.L., Coello, C.A.C., "Cultured differential evolution for constrained optimization", *Comput. Methods Appl. Mech. Engrg., vol.* 195, pp. 4303–4322, 2006.
21. Mezura-Montes, E., et al., "Simple Feasibility Rules and Differential Evolution for Constrained Optimization", *Proceedings of the 3rd Mexican International Conference on Artificial Intelligence (MICAI 2004),* 2004.
22. Liu, C.L., "Prediction Cash Flow for Construction Projects using Evolutionary Fuzzy Neural Inference Model", *MS Thesis, Department of Construction Engineering,* National Taiwan University of Science and Technology, Taipei, Taiwan, 2006.