Computers, Materials & Continua
DOI: 10.32604/cmc.2023.030818
Article

Project Assessment in Offshore Software Maintenance Outsourcing Using Deep Extreme Learning Machines

Atif Ikram[1,2,*], Masita Abdul Jalil[1], Amir Bin Ngah[1], Saqib Raza[6], Ahmad Salman Khan[3], Yasir Mahmood[3,4], Nazri Kama[4], Azri Azmi[4] and Assad Alzayed[5]

[1]Faculty of Ocean Engineering Technology and Informatics, University Malaysia Terengganu, Kuala Nerus, Malaysia
[2]Department of Computer Science & Information Technology, Faculty of Information Technology, The University of Lahore, Lahore, 54000, Pakistan
[3]Department of Software Engineering, Faculty of Information Technology, The University of Lahore, Lahore, 54000, Pakistan
[4]Advanced Informatics Department, Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, 54100, Kuala Lumpur, Malaysia
[5]Computer Science and Information Systems Department, College of Business Studies Public Authority for Applied Education and Training, (PAAET), Adailiyah, Kuwait
[6]Department of Computer Science, Sharif College of Engineering & Technology, Lahore, 54000, Pakistan
*Corresponding Author: Atif Ikram. Email: atif.ikram@cs.uol.ed.pk
Received: 02 April 2022; Accepted: 13 June 2022

Abstract: Software maintenance is the process of fixing, modifying, and improving software deliverables after they are delivered to the client. Clients can benefit from offshore software maintenance outsourcing (OSMO) in different ways, including time savings, cost savings, and improving the software quality and value. One of the hardest challenges for the OSMO vendor is to choose a suitable project among several clients' projects. The goal of the current study is to recommend a machine learning-based decision support system that OSMO vendors can utilize to forecast or assess the project of OSMO clients. The projects belong to OSMO vendors, having offices in developing countries while providing services to developed countries. In the current study, Extreme Learning Machine's (ELM's) variant called Deep Extreme Learning Machines (DELMs) is used. A novel dataset consisting of 195 projects data is proposed to train the model and to evaluate the overall efficiency of the proposed model. The proposed DELM's based model evaluations achieved 90.017% training accuracy having a value with 1.412 $\times$ 10^{-3} Root Mean Square Error (RMSE) and 85.772% testing accuracy with 1.569 $\times$ 10^{-3} RMSE with five DELMs hidden layers. The results express that the suggested model has gained a notable recognition rate in comparison to any previous studies. The current study also concludes DELMs as the most applicable and useful technique for OSMO client's project assessment.

Keywords: Software outsourcing; deep extreme learning machine (DELM); machine learning (ML); extreme learning machine; assessment

1 Introduction

Software maintenance spends more than 70% of the overall allotted budget for the software development lifecycle, giving it the longest lifespan and highest budget use. Software firms are increasingly outsourcing maintenance to attain quality while saving time and money. The OSMO vendor offers essential software maintenance facilities to OSMO clients anywhere in the world [1]. Even if the client's project is expected to generate revenue, the offshore services providing vendor should carefully consider the OSMO client's project since, as the adage goes, "everything that glitters is not gold." The problem lies in forecasting the parameters on the preliminary levels of project selection, whilst barriers of each initiative want to be mounted and whilst uncertainty concerning functionalities of OSMO. If the OSMO client's project is not properly assessed, it can significantly hinder the delivery of defined timeframes, budgets, and acceptable quality project deliverables. Oftentimes, confined understanding about influencing elements and threats which may also occur, stress from OSMO clients, and legacy software program estimation strategies primarily based totally on professional judgment may also result in obscure and typically over-optimistic estimates. Therefore, with the use of some assessment-related procedures, the OSMO vendor can choose a more suitable or appropriate project from several choices. Before accepting a client's project, it's critical to appraise, assess, or estimate it. So, there are several research papers on this topic that use estimation or machine learning techniques [2–5].

Machine learning (ML) is a field of artificial intelligence that uses computational and mathematical techniques. It is widely credited with the development of ' intelligent algorithms, which may be used to train computer systems using prior information or datasets to make intelligent judgments or predictions. Machine learning recognizes two essential artificial intelligence questions: How the program has been improved and what are the basic axioms of computer intelligence [6]. Since the development and emergence of the big data era, ML has had a key influence on many scientific disciplines and various applications [7,8]. Traditional machine learning is categorized into unsupervised, semi-supervised, supervised, and reinforcement learning [9]. The most decisive and challenging combination of ML and pattern recognition, known as supervised learning (SL) starts with an example of inputs and outputs [9,10].

Artificial Neural Networks (ANNs) are supervised learning-based exclusive data processing models for solving complex problems and understanding the behavior of complex systems using computer simulations. The definitive goal of the ANN algorithm is to handle and solve any computational query in the same way as any human brain handles the problem. An ANN is made up of many small processing units called neurons. Learning in ANN is the process of finding and adjusting weights. Analyzing the activity of neurons may necessitate a non-linear set of computing stages, each of which modifies the network's cumulative activation. The goal of ANN-based approaches is to expressly arraign credit beyond several of these phases [11]. ANN models such as the multi-layer perceptron and others have recently been used in research that has shown significant efficiency for a variety of prediction tasks [12].

Deep Learning (DL) is the extension of ANN. The initial principle of deep learning is to use unsupervised learning to first train each network layer [13]. DL approach is widely used to solve conventional and classic artificial intelligence challenges, and it has been applied to a broad variety of other domains and areas like prediction & classification [13]. DL comprises a wide range of supervised and unsupervised learning techniques and algorithms, encompassing artificial neural networks, hierarch probabilistic models, and several other computational methods [14]. The structure of deep learning is represented in Fig. 1.

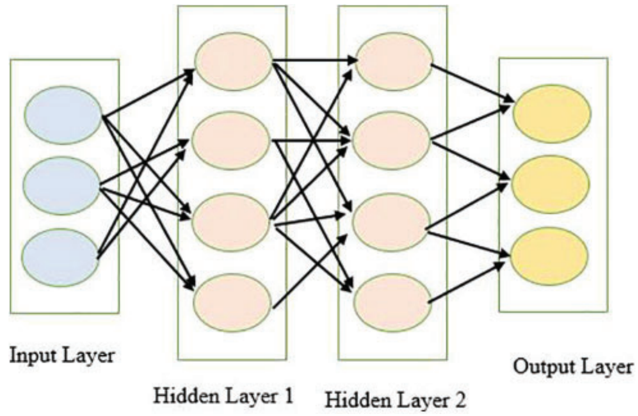

Figure 1: Structure of deep learning with multi layers [15]

Deep learning and classical machine learning approaches laid a solid basis for Extreme Learning Machines, a new efficient learning strategy established in the recent decade (ELM) [16]. Huang et al. originally proposed Extreme Learning Machines (ELM), a supervised learning technique. The key difference between ELMs and traditional Deep Learning (DL), is the bias of extreme learning machines and the development of random input weights. ELM has a very basic framework and computing operations with a restricted set of parameters; as a result of this potential characteristic, ELM has a faster learning network time training and enhanced generalization ability [17]. The ELM structure is presented in Fig. 2.

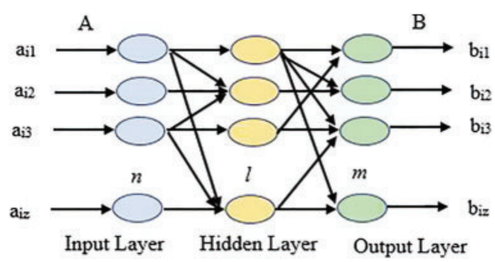

Figure 2: ELMs network diagram with m Output Neurons, n Input Neurons, and l Hidden Neurons [15]

The study [18] emphasizes that vendors should apply machine learning techniques for the assessment of the customer's project at the initial selection phase of the project lifecycle. The study finds that the use of ML methods will assist and help the manager in selecting the best project from a plethora of possibilities. The studies [19] and [20] mention different factors which can influence project selection. Although the studies are helpful to the software outsourcing industry, they have more emphasis on software development as compared to software maintenance outsourcing.

The current study will evaluate the OSMO client's project using Deep Extreme Learning Machines. A detailed discussion on Deep Extreme Learning Machines has been given in the next section. The current study has the following research questions:

RQ1: What are the most important attributes that may influence an OSMO client's project proposal?

RQ2: Which DL technique is best for predicting the OSMO client's project?

The aim of this research is to propose such an intelligent system which helps the OSMO vendor to take the decision while the selection of an appropriate project among many options. The prime motivation and goal behind this intelligent system is to manage and mitigate the risk involved in the appropriate project selection for OSMO vendor.

The organization of the remaining paper is like this: Section 2 highlights the existing literature. The proposed solution is discussed in Section 3. Section 4 is about the obtained results, 5 discussion and in 6 contains conclusion of the study.

2 Literature Review

During the early stages of a client's project selection, the OSMO vendor must have complete knowledge of the client's project. The trouble lies in the early stage of the right project selection. As a result, the wrong selection of project may rigorously impact the project deliverables, especially in the context of OSMO. Machine learning methods and algorithms are commonly used for the prediction, assessment, and right selection of software projects. The study [21] is fully related to software maintenance outsourcing but the researcher has not mentioned the offshore context in outsourcing. The study has presented such activities which can be helpful for SMO vendors while accepting the client's project. The future work of the study has motivated the researchers to implement any suitable machine learning or statistical techniques in the domain of software maintenance outsourcing.

The study [22] uses machine learning algorithms in software estimation so that managers may have precise predictions and the company can achieve sales targets effectively. The study has used multi-layer perceptron (MLP) and long short-term memory (LSTM) machine algorithms for the prediction process. The study has used a dataset of 77 projects. Although, this study is a good example to use machine learning algorithms for software predictions this study does not cover prediction in the OSMO context. The study [22] has admired that machine learning techniques can predict more accurately. As a result, software managers have a better idea about the upcoming project and they can do better planning to meet deadlines on time. The study [23] has recognized numerous factors that may be used as casual agents to assess the project of software development outsourcing. The study [23] has a worthy impact on the software outsourcing domain but with some limitations. These studies have not focused on software maintenance outsourcing rather the study focused on the domain of software development outsourcing. The study [24] has emphasized the significance of prediction in the early stages of software outsourcing. The study discussed that the maintenance outsourcing vendor has a very small amount of information at the project selection stage. So, predicting about offshore client's project offer would be highly beneficial for such a vendor. The study [24] has missing offshore context and focused only on Cobol/CICS based developed applications. The study [25] has also encouraged assessing the contracts to make them smart and to eliminate blind spots in contracts. Although the study [25] has discussed contracts but not covered the software maintenance outsourcing context. The study [26] has used Optimized Extreme Learning Machine (OELM) for the assessment of software maintenance-based projects. The study has compared the performance of OLEM with four different ML algorithms like Bagged CART, AdaBoost, Penalized Multinomial Regression, and Flexible Discriminant Analysis. The result of this comparison advocates the effective use of OELM in the field of software maintenance-based projects. The use of AI in different fields is evident [27–30].

Tab. 1 gives an overview of the most relevant techniques used in the context of this research. Hence the above-given literature and Tab. 1 show the importance of the implementation of AI techniques for the software industry. Therefore, we have selected DELM for project assessment in the domain of software maintenance outsourcing.

Table 1: Techniques used in the assessment of software projects

Study ID	Objective/Claim	Techniques used	Dataset	Conclusion
[4]	Effective project selection is possible with the help of ML techniques	An artificial neural network, MLP (Multilayer Perceptron)	Primary sources: 10 interviews Secondary sources: 150 projects	The ANN (MLP) model helps to assess and early evaluation of critical success factors/attributes.
[22]	To facilitate project managers to meet the user requirements within the specific given time.	LSTM and MLP	With a dataset of 77 projects	The results of MLP (Multilayer Perceptron) have a better performance as compared to LSTM (Long Short-Term Memory).
[24]	To facilitate software maintenance outsourcing vendor to predict about required maintenance efforts, even when very little information is available.	Code Metrics	55 applications	Techniques like code metrics can be used to predict software maintenance efforts
[31]	Estimation of software maintenance efforts the absence of some key parameters like maintenance history and development effort	Taguchi's method (Statistical Analysis)	6 experts for interview	Statistical Analysis methods like Taguchi's method can be used to estimate maintenance efforts
[32]	To check the application of Neural Network (NN) in software maintenance effort estimation	Neural Network (Bayesian regularization training and algorithm named as back-propagation), using Matlab tool	36 projects	The Neural Network-based estimator can be effectively used (with Matlab tool) for software maintenance estimation
[33]	Presented an effective approach to estimate project duration and effort during software projects	Average of Support Vector Machines, Generalized Linear Models and Neural Networks	1192 projects	The implementation of ML algorithms provides better results as matched to other estimation techniques.
[34]	Effective use of the pre-processing techniques on machine learning techniques	case-based reasoning, artificial neural networks, regression trees and classification	14 features from 6000 projects (ISBSG dataset)	Only pre-processing techniques are not important, But the right selection of techniques in the context of ML methods and datasets are also important.

(Continued)

Table 1: Continued

Study ID	Objective/Claim	Techniques used	Dataset	Conclusion
[35]	Empirically validate that behavior of multi- types of NN in effort estimation of Agile based large software, using SPA (story point approach)	Cascade-Correlation Neural Network, Polynomial Neural Network, General Regression Neural Network, Probabilistic Neural Network	21 Projects	It is empirically validated that the cascade network performs better.
[36]	Enhancing software estimation accuracy with the help of ANN (artificial neural network)	Multilayered Feed Forward Neural Network (MFNN)	13 projects	As artificial neural network allows any quantity of input values, so artificial neural network provides a big advantage to the software estimation field
[37]	In software effort estimation frameworks, compare the "One-size-fit-all" approach Vs Customizable attribute selection	Neuro-Fuzzy Inference System (ANFIS training, NN training), Expert Judgment	6 Projects	'Neuro-Fuzzy Inference System' along with 'customized attribute selection' and 'expert judgment' can be successfully used for software estimations.
[38]	To examine the factors contributing to the success of offshore software projects	Partial Least squares (PLSs)	47 offshore experts	Partial Least Squares (PLSs) can be used to find the relationship between factors and the success of offshore software projects

3 Proposed Solution

The model proposed for the DELM-based model is shown in Fig. 3. The detail of every step is discussed further in this section.

3.1 Initial Analysis

This preliminary analysis indicated some fundamental independent variables (Tab. 2) which led towards providing an intelligent and optimized deep learning-based solution for better predictions of OSMO proposals. This multivariate analysis deals with the statistical analysis of data collected on more than one dependent variable. Multivariate techniques are popular because they help organizations to turn data into knowledge and thereby improve their decision-making.

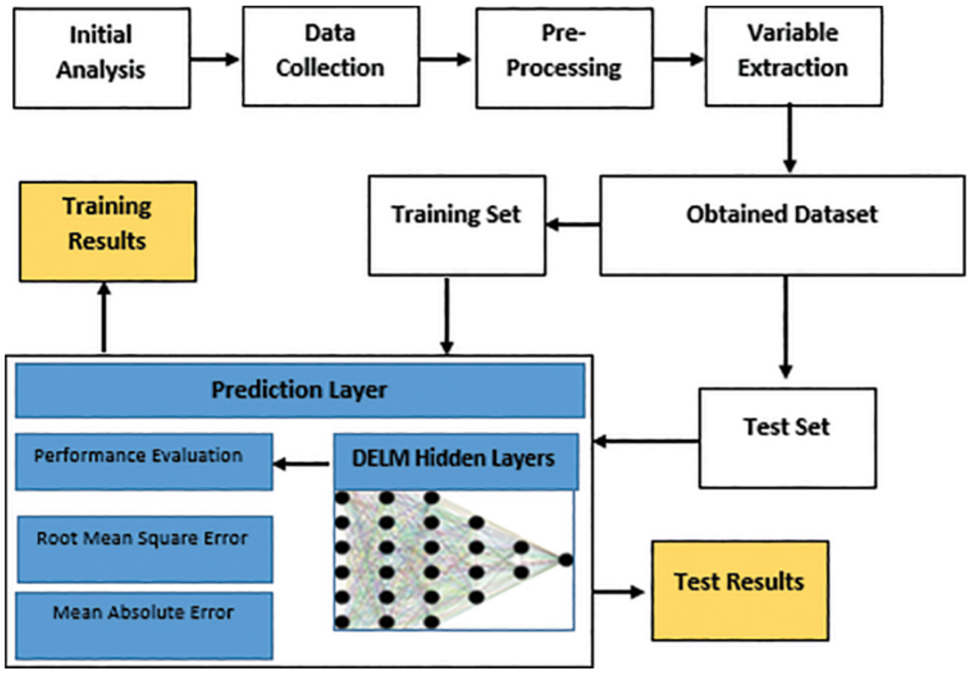

Figure 3: Deep extreme learning machine based proposed model

Table 2: Selected variables for client's project assessment

Sr#	Variable	Description	Nominal Scale
1	Size of the supplier organization	Number of employees a vendor has in the organization.	Small, medium, large
2	Required team size	Number of employees required by the client	Small, medium, large
3	Domain of the project	Project belongs to which domain	Same, partial same, new domain
4	Size of software maintenance project	Size of the client project	Small, medium, large
5	Use of international standards in the software development	What kind of standard clients have followed in project development?	Fully, Partially, no,
6	Software code complexity	Difficulty to understand the code	Easy, fair, challenging

(Continued)

Table 2: Continued

Sr#	Variable	Description	Nominal Scale
7	Use of international standards in software documentation	What kind of standard clients have followed in the documentation of the project	Fully, partially, no
8	Quality of related document	Quality of design document, user manual, etc	Good, average, bad
9	Required type of maintenance	Maintenance required by the client	Single, multiple
10	Structure of code	What kind of standard clients have used during software design?	Good, average, poor
11	Common time zone	Common working hours between vendor and client	Partially, fully
12	Client's market reputation	What kind of reputation client has?	Good, Avg, Bad
13	Handover experience of the client	It determines the experience of the client in the transition of software.	Fully, partially, no_experience
14	Operating language	(client's language) (similar or other)	English as a common language
15	Nature of SLA	Service Level Agreement	Fair, unclear, biased
16	Methodology adopted	Methodology adopted during software development	Waterfall, Iterative, agile
17	System Age	How much time has been passed after software development?	Newly developed, Legacy system

3.2 Data Collection

The data collection process consists of collecting and measuring information about variables of interest in an established systematic relationship that allows you to answer research questions, test hypotheses, and evaluate results. The multivariate analysis deals with the statistical analysis of data collected on more than one dependent variable. Multivariate techniques are popular because they help organizations to turn data into knowledge and thereby improve their decision-making.

In this phase of the study, a questionnaire is prepared to frame collected data on the dependent and independent variables of the study. The questionnaire was based on structured and unstructured questions. The questionnaire was sent to 195 software development and related services providing companies of the countries like India, China, Pakistan and Bangladesh. In the reply, the current research received data from 483 software maintenance outsourcing projects.

3.3 Pre-processing

To improve the accuracy of the proposed model, the training set must be complete, continuous, and noiseless. Pre-processing is a process of inspecting, cleansing, transforming, and modeling data. The response recorded from different companies had certain major issues like missing values, redundancy, etc. After the pre-processing phase, 455 software maintenance outsourcing projects responses were considered for the further phases of the study.

3.4 Variable Extraction

The detail of the variables extracted from the collected data with description is illustrated in Tab. 2.

3.5 Obtained Dataset

The obtained dataset consists of 455 instances with 17 nominal attributes where each instance belongs to one label class. For this study, the dataset of 455 instances is divided into 70% of the training set (333 instances) and 30% testing set (122 instances). The dataset instances are categorized into three label classes: Reject, Risk, and Accept. The detailed description of attributes and classes is already discussed in Tab. 2. Few dataset instances are illustrated here as a sample where 'Accept', 'Risk' and 'Reject' are label classes:

medium, small, partial_same, medium, partially, easy, partially, avg, multiple, avg, partially, good_customer,Fully,English_as_Common,Fair,Waterfall,Legacy, Accept

medium, small,Same, large, no, challenging, fully, good, single, avg, fully,Avg_customer, partially, English_as_Common,Unclear ,Iterative, Legacy, Risk

medium, large, new_domain, large, partially, easy, fully, good, multiple, good, fully, bad_customer, no_experience,Language_barrier,Common, unclear,Agile,Newely_dev, Reject.

3.6 Prediction Layer

The Deep Extreme Learning Machines (DELMs) is a well-known technique used in various areas for image classification. The traditional ANN algorithms require more samples and slow learning times and can over-fit the learning model. The DELM is used widely in various areas for classification and regression purposes because DELM learns fast and it is efficient in terms of cost of computational complexity. The three types of layers included in the DELM model are the input layer, multiple hidden layers, and the output layer. The structural model of a DELM can be seen in Fig. 4.

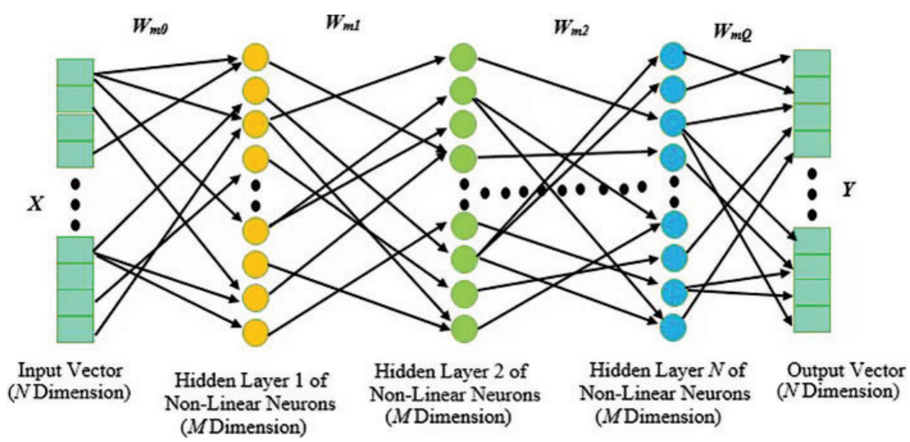

Figure 4: Structure of deep extreme learning machines, N-dimensional input vector X is devised to hide layers M dimensional by using a random weight matrix [36]

After the transformation of sigmoid hidden units, the result is multiplied by output weights to construct vector Y of similar dimensions as an input. Vector Y is further devised to M dimensional second layer using another matrix with random weights Wm2. Using training data, the output matrix having weight Wm0 is obtained by solving Eq. (1).

$$VX^T = W_{m0}\left(XX^T + R_J\right) \quad (1)$$

where R is the regularization parameter.

First consider {X, Y} = {X_i, Y_i} where (k = 1, 2, 3 . . . , n) and there is an input feature X = [x_{i1}, x_{i2}, x_{i3} . . . , x_{iQ}] and desired matrix Y = [y_{j1}, y_{j2}, y_{j3} . . . , y_{jQ}]

$$X = \begin{bmatrix} x_{11} & x_{12} \ldots, & x_{1Q} \\ x_{21} & x_{22} \ldots, x_{2Q} & \\ & & \\ x_{n1} & x_{n2} \ldots, x_{nQ} & \end{bmatrix}$$

$$Y = \begin{bmatrix} y_{11} & y_{12} \ldots, & y_{1Q} \\ y_{21} & y_{22} \ldots, y_{2Q} & \\ & & \\ y_{m1} & y_{m2} \ldots y_{2Q} & \end{bmatrix} \tag{2}$$

$$W = \begin{bmatrix} w_{11} & w_{12} \ldots, & w_{1Q} \\ w_{21} & w_{22} \ldots, w_{2Q} & \\ & & \\ w_{l1} & w_{l2} \ldots, w_{lQ} & \end{bmatrix} \tag{3}$$

where W is the weight between input layers and hidden layers.

$$\mu = \begin{bmatrix} \mu_{11} & \mu_{12} \ldots, \mu_{1z} \\ \mu_{21} & \mu_{22} \ldots, \mu_{2Z} \\ & \\ \mu_{lz} & \mu_{lZ} \ldots, \mu_{lZ} \end{bmatrix} \tag{4}$$

where μ lz expresses weights amongst nth output-layer neurons and mth hidden-layer of neurons.

$$\boldsymbol{Y} = [y_1, \ y_2, \ y_3 \ldots, \ y_n]^T \tag{5}$$

The output matrix L can be represented by:

$$\boldsymbol{L} = [l_1, \ l_1, \ l_3 \ldots, \ l_n]_{m*Q} \tag{6}$$

for each column output of matrix L expressed as follow:

$$\boldsymbol{l}_j = \begin{bmatrix} l_{1j} \\ l_{2j} \\ \\ l_{mj} \end{bmatrix} = \begin{bmatrix} \sum_1^l \mu_{i1} g \ \left(w_i x_j + b_i\right) \\ \sum_1^l \mu_{i2} g \ \left(w_i x_j + b_i\right) \\ \\ \sum_1^l \mu_{im} g \ \left(w_i x_j + b_i\right) \end{bmatrix} \tag{7}$$

$(j = 1, 2, 3, \ldots Z)$

Consider (5) and (6) we get

$$\boldsymbol{L}' = H\mu \tag{8}$$

The transpose of L is denoted by L′ and H denotes the hidden layer’s output.

$$\mu = H + \boldsymbol{L}' \tag{9}$$

The regularization term is added to μ.

$$H_1 = \boldsymbol{L}\mu^{-1} \tag{10}$$

$\mu-1$ is an inverse of the matrix μ.

$$H_1 = g\,(WH_1 + \mu_1) \tag{11}$$

an estimated second hidden-layer input is expressed.

$$W_{HE} = g^{-1}\,(H_1)\,H_E{}^{+} \tag{12}$$

The outcome of second hidden layer is revised as, by indicating the correct $f(x)$ activation function:

$$H_2 = f\,(W_{HE}\ H_E) \qquad \text{Where } \mathrm{W_{HE}H_E} = \mathrm{NEth_2}$$

$$\mathrm{H}_2 = f\,(N_E th_2) \tag{13}$$

The estimated layer 3 results are shown in Eq. (15).

$$\mu_{new} = H_2 N^{+} \tag{14}$$

$$H_3 = \mu^{+}{}_{new} \tag{15}$$

The output of the third layer as shown in Eq. (16).

$$\mu_{new} = H_2{}^{+} N \tag{16}$$

$$H_3 = N\mu^{+}{}_{new} \tag{17}$$

Eqs. (9) and (10) allows the third layer output to be

$$H_2 = g^{-1}\,(H_2 W_2) = g\ \ (N_E th_2) \tag{18}$$

$$W_{HE1} = \mu^{-1}\,(H_2)\,H_{EI}{}^{+} \tag{19}$$

In Eq. (21), third hidden layer output is calculated as:

$$g\,(x) = \frac{1}{1 + e^{-x}}$$

$$H_2 = g\,(W_{HE1} H_{E1}) \ \text{ Where } \mathrm{W_{HE1}H_E} = \mathrm{N_E th_2} \tag{20}$$

$$H_2 = g\,(N_E th_2) \tag{21}$$

The likely outcome of level 3 hidden layer.

$$\mu_{new} = H_4{}^{t}\left(\frac{1}{\lambda} + H_4{}^{t}\ H_4\right)^{-1} N \tag{22}$$

$$H_4 = N\mu^{+}{}_{new} \tag{23}$$

The transposed weight matrix μ_{new} is $N\mu^{+}{}_{new}$. After that Deep Extreme Learning Machines (DELM) creates the matrix $W_{HE2} = [B_3, W_3]$. Eqs. (13) and (24) can be utilized to achieve the output of the hidden-layer number fourth.

$$H_4 = g^{-1}\,(H_3 W_3 + B_3) = g\left(N_E th_{4,1}\right) \tag{24}$$

$$W_{ME2} = \mu^{-1}\left((H_4)\, M^{+}{}_{Q2}\right) \tag{25}$$

The sigmoidal logistic function is used in Eq. (26), following is the measuring of 3rd and 4th hidden layer values.

$$H_4 = g\left(Neth_{4,2}\right) \tag{26}$$

Eq. (29) shows the required output of the DELM system.

$$\beta_{new} = H_{nt}{}^{t}\left(\frac{1}{\lambda} + H_{nth}{}^{t}\, H_{nth}\right)^{-1} N \tag{27}$$

$$M_{nth} = N\mu^{+}{}_{new} \tag{28}$$

$$f\,(x) = m_{nth}\mu_{new} \tag{29}$$

The last result of the Deep Extreme Learning Machines (DELM) network is computed. If hidden layers are increased, the output of other hidden layers will be calculated with the same technique.

$$op = \frac{1}{1 + e^{-Nethj}} \quad \text{Where j} = 1, 2, 3 \ldots, \text{r} \tag{30}$$

4 Obtained Results

In this study, the OSMO client proposal is measured with two basic parameters which are accuracy and miss rate. The accuracy and miss rate of the proposed DELMs system is evaluated by Eq. (31) and Eq. (32) respectively. The accuracy, miss rate, and Root Mean Square Error (RMSE), training time, and testing time obtained from the training results and testing results, with the different number of DELMs hidden layers used in the prediction phase are expressed in Tab. 3.

$$\text{Accuracy} = \frac{\text{No of Correctly Classified Instances}}{\text{Total No of Instances}} * 100 \tag{31}$$

$$\text{Miss Rate} = \frac{\text{No of Incorrect Correctly Classified Instances}}{\text{Total No of Instances}} * 100 \tag{32}$$

Table 3: Performance of the proposed DELM based model

No. of Hidden Layers	Training Results				Test Results			
	Accuracy (%)	Miss Rate(%)	Time (mili-sec)	RMSE	Accuracy (%)	Miss Rate (%)	Time (mili-sec)	RMSE
3	88.362	12.634	5.124	1.512×10^{-3}	84.822	15.116	3.167	1.613×10^{-3}
4	89.625	11.374	5.230	1.488×10^{-3}	85.221	14.781	3.238	1.601×10^{-3}
5	**90.017**	**9.073**	**5.317**	$\mathbf{1.412 \times 10^{-3}}$	**85.772**	**14.290**	**3.482**	$\mathbf{1.591 \times 10^{-3}}$
6	90.189	9.811	5.545	1.401×10^{-3}	85.776	14.224	3.558	1.569×10^{-3}

5 Discussion

The simulations were performed on 6th generation Core i5–6200U 2.3 GHz with 8 GB RAM. MATLAB 9.4 R2018a tool is used for simulating the results, which generate a complexity of the scheme in terms of time execution (milliseconds). The recognition rate by the proposed DELM-based system achieved 90.017% training with 1.412×10^{-3} RMSE and 85.772% testing results with 1.591×10^{-3} RMSE, with five hidden layers. By the introduction of the sixth hidden layer, only training results are improving. The system accuracy is recognized by the test results, which are not improving with the sixth hidden layer. With the introduction of the sixth hidden layer, the computational complexity is increasing.

WEKA™ version 3.8 has been used to find the efficiency of the proposed technique and the dataset by using different supervised learning based classifiers. Different categories of classifiers and algorithms are available for SL, like Logistics, Naïve Bayesian, Lazy, IBK and Tree etc.

The comparison of the proposed DELM based model with other machine learning techniques using the same proposed dataset is illustrated in Tab. 4. It is evident from the results that DELM performs better as compared to three other ML-based techniques.

Table 4: Comparison of DELM with different machine learning classifiers

Machine Learning Technique	Training Results	Testing Results
Naïve Bayesian	81.565%	77.465%
Logistics	77.623%	72.234%
IBK	71.621%	68.563%
Deep Extreme Learning Machine (DELM)	90.017%	85.772%

Note: The testing accuracy achieved by different SL classifiers like Naïve Bayes, Logistics, and IBK reached up to 77.465%, 72.234%, 68.563% respectively. The obtained results elaborates the effectiveness of the DELM based proposed technique.

6 Conclusion

The selection of an appropriate project, especially for OSMO vendors, is a complex and risky phenomenon. Consequently, it is required to propose such a mechanism that can help OSMO vendors in the selection of an appropriate project. This study proposes a DELM-based client project assessment decision support system. The proposed model will facilitate the OSMO vendor to select an appropriate project amongst many projects. A recently proposed ELM-based algorithm with multi-hidden layers, random weights, and bias called DELMs is used in the assessment phase. 90.017% training with 1.412×10^{-3} RMSE and 85.772% testing results with 1.591×10^{-3} RMSE, with five hidden layers. The obtained results illustrate the efficiency of the overall model which supports the OSMO vendor to select an appropriate project amongst different options. This study also concludes the efficiency of DELMs (as shown in Tab. 4) for the assessment of the OSMO client project. Hence, the results proved that DELMs gave better results in the comparison of the other three techniques. The proposed study has considered seventeen variable for the client project assessment. There can be considered more independent variables for future studies.

Acknowledgement: The authors would like to thank and acknowledge to Universiti Malaysia Terengganu (UMT) Malaysia, Muhammad Akram from Department of Computer Science, College of Computer Science and Information Systems, Najran University, Saudi Arabia, Computer Science and Information Systems Department at Public Authority for Applied Education and Training, (PAAET) Kuwait and every individual who has been a source of information, support and encouragement on successful completion of this manuscript.

Funding Statement: This research is fully funded by Universiti Teknologi Malaysia under the UTM Fundamental Research Grant (UTMFR) with Cost Center No Q.K130000.2556.21H14.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Ikram, M. A. Jalil, A. B. Ngah and A. S. Khan, "Towards offshore software maintenance outsourcing process model," *International Journal of Computer Science and Network Security*, vol. 20, no. 4, pp. 6–14, 2020.
[2] H. Alsolai and M. Roper, "A systematic literature review of machine learning techniques for software maintainability prediction," *Information and Software Technology*, vol. 119, pp. 106214, 2020.
[3] X. Sheng and J. Lu, "The application of machine learning in determining earthquake magnitude as an early warning," *IOP Conference Series: Earth and Environmental Science*, vol. 783, no. 1, 2021.
[4] F. Costantino, G. D. Gravio and F. Nonino, "Project selection in project portfolio management: An artificial neural network model based on critical success factors," *International Journal of Project Management*, vol. 33, no. 8, pp. 1744–1754, 2015.
[5] S. Ali, H. Li, S. U. Khan, M. F. Abrar and Y. Zhao, "Practitioner's view of barriers to software outsourcing partnership formation: An empirical exploration," *Journal of Software: Evolution and Process*, vol. 32, no. 5, 2020.
[6] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science Special Edition Artificial Intelligence*, vol. 349, no. 6245, pp. 255–260, 2015.
[7] L. Zhou, S. Pan, J. Wang and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, 2017.
[8] O. Y. Al-Jarrah, P. D. Yoo, S. Muhidat, G. K. Karagiannidis and K. Taha, " "Efficient machine learning for big data: A review," *Big Data Research*, vol. 2, no. 3, pp. 87–93, 2015.
[9] F. Schwenker and E. Trentin, "Pattern classification and clustering: A review of partially supervised learning approaches," *Pattern Recognition Letters*, vol. 37, pp. 4–14, 2014.
[10] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed, S. H. Shirazi *et al.,* "Offline cursive urdu-nastaliq script recognition using multidimensional recurrent neural networks," *Neurocomputing*, vol. 177, pp. 228–241, 2016.
[11] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
[12] M. Fayaz and D. Kim, "A prediction methodology of energy consumption based on deep extreme learning machine and comparative analysis in residential buildings," *Electronics*, vol. 7, no. 10, pp. 1–22, 2017.
[13] Y. Gu, Y. Chen, J. Liu and X. Jian, "Semi-supervised deep extreme learning machine for Wi-fi based localization," *Neurocomputing*, vol. 166, pp. 282–293, 2015.
[14] A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, pp. 1–13, 2018.
[15] S. S. R. Rizvi, A. Sagheer, K. Adnan and A. Muhammad, "Optical character recognition system for nastalique urdu-like script languages using supervised learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 10, pp. 1–32, 2019.

[16] L. Yao and Z. Ge, "Deep learning of semi-supervised process data with hierarchical extreme learning machine and soft sensor application," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1–8, 2017.
[17] G. B. Huang, Q. Y. Zhu and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2005.
[18] S. F. Alhashmi, S. A. Salloum and S. Abdallah, "Critical success factors for implementing artificial intelligence (AI) projects in Dubai government United Arab Emirates (UAE) health sector: Applying the extended technology acceptance model (TAM)," in *Int. Conf. on Advanced Intelligent Systems and Informatics*, Cairo, Egypt, pp. 393–405, 2019.
[19] S. U. Khan, M. Niazi and R. Ahmad, "Factors influencing clients in the selection of offshore software outsourcing vendors: An exploratory study using a systematic literature review," *Journal of Systems and Software*, vol. 84, no. 4, pp. 686–699, 2011.
[20] H. U. Rahman, M. Raza, P. Afsar, H. U. Khan and S. Nazir, "Analyzing factors that influence offshore outsourcing decision of application maintenance," *IEEE Access*, vol. 8, pp. 183913–183926, 2020.
[21] A. Ikram, H. Riaz and A. S. Khan, "Eliciting theory of software maintenance outsourcing process: A systematic literature review," *International Journal of Computer Science and Network Security*, vol. 18, no. 4, pp. 132–143, 2018.
[22] E. F. Predescu, A. Tefan and A. V. Zaharia, "Software effort estimation using multilayer perceptron and long short term memory," *Informatica Economica*, vol. 23, no. 2, pp. 76–87, 2019.
[23] S. U. Khan and A. W. Khan, "Critical challenges in managing offshore software development outsourcing contract from vendors' perspectives," *IET Software*, vol. 11, no. 1, pp. 1–11, 2017.
[24] M. Polo, M. Piattini and F. Ruiz, "Using code metrics to predict maintenance of legacy programs: A case study," in *Proc. IEEE Int. Conf. on Software Maintenance*, Florence, Italy, pp. 202–208, 2001.
[25] R. Unsworth, "Smart contract this! an assessment of the contractual landscape and the herculean challenges it currently presents for "self-executing" contracts," *Legal Tech, Smart Contracts and Blockchain*, pp. 17–61, 2019.
[26] S. Gupta and C. Anuradha, "An optimized extreme learning machine algorithm for improving software maintainability prediction," in *11th Int. Conf. on Cloud Computing, Data Science & Engineering*, Uttar Pradesh, India, pp. 829–836, 2021.
[27] M. Ashraf, F. Ahmad, R. Rauqir, F. Abid, M. Naseer *et al.,* "Emotion recognition based on musical instrument using deep neural network," in *Int. Conf. on Frontiers of Information Technology*, Islamabad, Pakistan, pp. 323–328, 2021.
[28] M. Ashraf, G. Geng, X. Wang, F. Ahmad and F. Abid, "A globally regularized joint neural architecture for music classification," *IEEE Access*, vol. 8, pp. 220980–220989, 2020.
[29] S. Basri, N. Kama, S. Adli and F. Haneem, "Using static and dynamic impact analysis for effort estimation," *IET Software*, vol. 10, no. 4, pp. 89–95, 2016.
[30] Y. Mahmood, N. Kama and A. Azmi, "A systematic review of studies on use case points and expert-based estimation of software development effort," *Journal of Software: Evolution and Process*, vol. 32, no. 7, 2020.
[31] B. S. Rao and N. L. Sarda, "Effort drivers in maintenance outsourcing-An experiment using taguchi's methodology," in *Seventh European Conf. on Software Maintenance and Reengineering*, Benevento, Italy, pp. 271–280, 2003.
[32] R. Shukla and A. K. Misra, "Estimating software maintenance effort: A neural network approach," in *Proc. of the 1st India Software Engineering Conf.*, Hyderabad, India, pp. 107–112, 2008.
[33] P. Pospieszny, B. Czarnacka-Chrobot and A. Kobylinski, "An effective approach for software project effort and duration estimation with machine learning algorithms," *Journal of Systems and Software*, vol. 137, pp. 184–196, 2018.
[34] J. Huang, Y. F. Li and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Information and Software Technology*, vol. 67, pp. 108–127, 2015.
[35] A. Panda, S. M. Satapathy and S. K. Rath, "Empirical validation of neural network models for agile software effort estimation based on story points," *Procedia Computer Science*, vol. 57, pp. 772–781, 2015.

[36] P. Rijwani and S. Jain, "Enhanced software effort estimation using multi layered feed forward artificial neural network technique," *Procedia Computer Science*, vol. 89, pp. 307–312, 2016.
[37] K. Grolinger, B. Muslimi, M. A. Capretz and M. Benko, "EEF-CAS: An effort estimation framework with customizable attribute selection," *International Journal of Advancements in Computing Technology*, vol. 5, no. 13, pp. 1–14, 2013.
[38] M. Westner and S. Strahringer, "Determinants of success in IS offshoring projects: Results from an empirical study of German companies," *Information & Management*, vol. 47, no. 5–6, pp. 291–299, 2010.