Journal of Scientific Research and Development 2 (7): 134-140, 2015
Available online at www.jsrad.org
ISSN 1115-7569

A new approach with hybrid of artificial neural network and k-nearest neighbor algorithms in cost estimation of CMS based web sites designing

Laya Ebrahimi Jahatloo [1], Ahmad Jafarian [2,*]

[1]*Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran*
[2]*Department of Mathematics, Urmia Branch, Islamic Azad University, Urmia, Iran*

Abstract: Increasing use of web applications makes web designers develop web pages with rich content. Cost and effort are two key factors in designing web projects which are predicted by a precise model. Most web cost estimation models are algorithmic that use a number of variables, known as cost drivers, calculate non-functional traits in an attempt to evaluate web development projects. These models formulate the relationship between effort and project characteristics and determine size as essential characteristics. In recent years, artificial intelligence models such as data mining techniques and artificial neural networks are proposed as an alternative to algorithmic and traditional methods. Artificial intelligence based methods compensates for the shortcomings of algorithmic models and yield better outcomes by applying data learning methods for prediction. The present study combines K-Nearest Neighbor (KNN) and ANNs Multi-Layer Perceptron (MLP) to present the proposed model for web cost estimation based on content management system.

Key words: Web cost estimation, Artificial Neural Networks; K-nearest neighbor; Content management system

1. Introduction

Today, wide use of the internet makes the web designing companies to pay excessive attention to content management system (CSM) which provides the users with various capabilities (Mo et al., 2014). CMS is an open source system that fully supports developing, managing and updating a web site and encompasses all necessary tools to manage a web site. CMS includes the life cycle of a page from development to expiration. It also enables the users to manage web site structure and style, and their relationships with menus. In other words, CMS is strong software for developing professional web sites.

CMS differs from traditional methods in terms of cost and design. In traditional methods, number of pages before and after ordering is limited and any additional page needs extra payment on the side of the customer. These restraints are evaluated as prominent parameters in traditional web estimation models (Costagliola et al., 2005; Mendes et al., 2007). However, CMS imposes no limitation on number of pages. On the other in hand traditional methods, effort and cost estimation is performed based on expertise and mastering programming languages. But, project members involved in the project is considerably lower in CMS because its open source nature and in-advance implementation. Also project members are encouraged to develop modules rather than to program. Generally, CMS based web sites focus on graphic, gallery, essential modules development and support instead of costs of programming, human resources, expertise, number of script lines and number of pages since these options are embedded in CMS open source packets (Seel, 2012; Mendes, 2014). CMS eliminates, optimizes, modifies or updates most of the parameters used in the traditional models of project cost estimation and adds new parameters to the existing ones. Here, a new hybrid model based on ANN MLP (German et al., 1992) and KNN (Gharehchopogh et al., 2015; Martin, 1995) is proposed to develop and estimate cost of web page designing projects. The proposed model exploits new technologies in web designing and predicts costs of web page designing for companies, developers and costumers. Since different factors and variables intervene in a project, precise techniques are to be applied to ensure higher prediction accuracy in early stages of the project. These techniques, as well as dataset characteristics embedded in them, affect prediction accuracy. The present study uses dataset of projects S.R.KHASE (Khaze et al., 2013) including 99 webpage projects. The study is organized as follows. Section two explores the related literature on WCS. Section three elaborates on the proposed model. Section four explained the results and evaluations. Finally in section five the concluding remarks are given.

2. Related works

A bulk of studies has been done on cost estimation of web projects but there is no general agreement on the best method for web developing

* Corresponding Author.

companies. Naïve Bayes algorithm is suggested for cost estimation CMS based web sites (Khaze et al., 2013). Evaluations are done on 99 web projects and the results given accordingly. The Classification accuracy and inaccurate prediction of classification are 55% and 45% respectively. Results indicate that the proposed model is effective for WCS. The WEBMO model, which is developed from COCOMO II, is an algorithmic model for effort estimation and prediction of effort rate of webpage projects (Reifer, 2000). The web object model is taken as our measure. It uses data from 64 web projects and expert comments and enables users to adjust costs by means of cost derivers and calculate actual cost by determining certain characteristics. This model uses web object estimators and complexity coefficient table for objects to initiate the process of calculating operators and operands of system and the web objects. Then, by determining measures of complexity, it estimates web costs in areas of e-business, financial and trade applications, task to task, portals and information services. WEBMO model differs from COCOMO II in that it has 9 cost drivers, rather than 7, and variable rather than constant power. Finally, PRED (n) standard is used in the model to analyze the proposed model.

Scholars (Mendes et al., 2001) proposed 5 distinct classifications for website measuring criteria, including hyper-media applications, web applications, web pages, media and programs. In web applications, factors of number of pages, number of media, number of programs, total page space, total media space, total code size, number of reused media, number of reused programs, total space for reused media, total reused code size, code description size, reused code description size, page complexity, links, number of links and cyclomatic complexity are evaluated. In web pages, factors of page allocation, page complexity, graphic complexity, audio complexity, image complexity, animation complexity, scanned image complexity and links complexity are evaluated. In the field of media, factors of media duration and allocation are considered. Finally, code size indicating total number of codes in a program is evaluated in the field of programs. Taking the above factors and 77 datasets, this model provides a linear regression prediction of new projects to estimate costs and efforts. The evaluation is validated using MMRE standard. Number of studies (Mendes et al., 2002) has relied on number of Use Cases in Use Case Chart, number of entities and pages in entity-relationship model, number of nodes and anchors in Navigation chart and time spent on web designing. Then, number of html pages, media files, total number of terms in JavaScript codes and cascade method, total number of internal and external links in each page, and number of media differences in each page are counted using a case oriented inference technique. In addition, data from 25 databases are used for cost and effort estimation of new projects. Various evaluating standards of the estimation model such as MMRE, MdMRE and PRED (25) are used.

Some case studies have investigated cost estimation of web based software projects (Mendes et al., 2002) using 73 databases, case based inference techniques, linear regression, and classification techniques such as decision trees and regression trees. Here, cost estimation is done by measuring number of pages, number of media, number of programs, and number of reused programs, links, density, and total page complexity. This model is also evaluated using evaluating standards of MMRE, MdMRE and PRED (25). Experts (Ruhe et al., 2003) proposed COBRA model to do a cost and effort estimation of web projects by data collected from small firms. COBRA is a method for developing logical cost estimation from data of a specific firm and employs expert ideas and data collected from previous projects to perform cost estimation. This model is evaluated using web objects in 12 finished projects. The researchers have suggested a model based on experts idea and linear regression prediction which is evaluated by standards of MMRE and PRED (25).

3. Proposed model

Today, websites play an undeniable and essential role in spreading scientific and educational materials. Many web content developing companies are established as a result of an increasing demand and use of websites. Accordingly website cost estimation is of utmost importance for the web developing companies and their stake holders. Thus, the present study combines MLP and KNN algorithms to improve cost estimation. The flowchart of the proposed model is illustrated in Fig. 1.

It is noteworthy that in ANN, the proposed model uses error post propagation algorithm and logistic activation function as activator function between neurons. As can be inferred from Fig. 1, results obtained from ANN MLP are first stored in the system and, then, directed along with test and training dataset to KNN algorithm. The KNN model is a method for classifying objects based on the nearest training sample technique in attribute space where all training samples are stored first and classification is delayed until an unknown sample demands classification (Gharehchopogh et al., 2015). Training samples are described in KNN in terms of numerical n-dimension attributes. Each sample is displayed by a point in an n-dimensional space. Therefore, all training samples are stored in an n-dimensional training pattern. In case an unknown sample demands classification, the algorithm searches the training pattern space for k-samples surrounding the unknown sample. This vicinity is defined by Euclidean distance. When 2 points include $X_1=(x_{11}, x_{12}, x_{13},....x_{1n})$ and $X_2=(x_{21},x_{22},x_{23},....x_{2n})$, Euclidean distance between them is calculated from Eq. (1).

$$d(O_1, O_2) = \sqrt{\sum_{j=1}^{k} (x_{1,j} - x_{2,j})^2} \qquad (1)$$

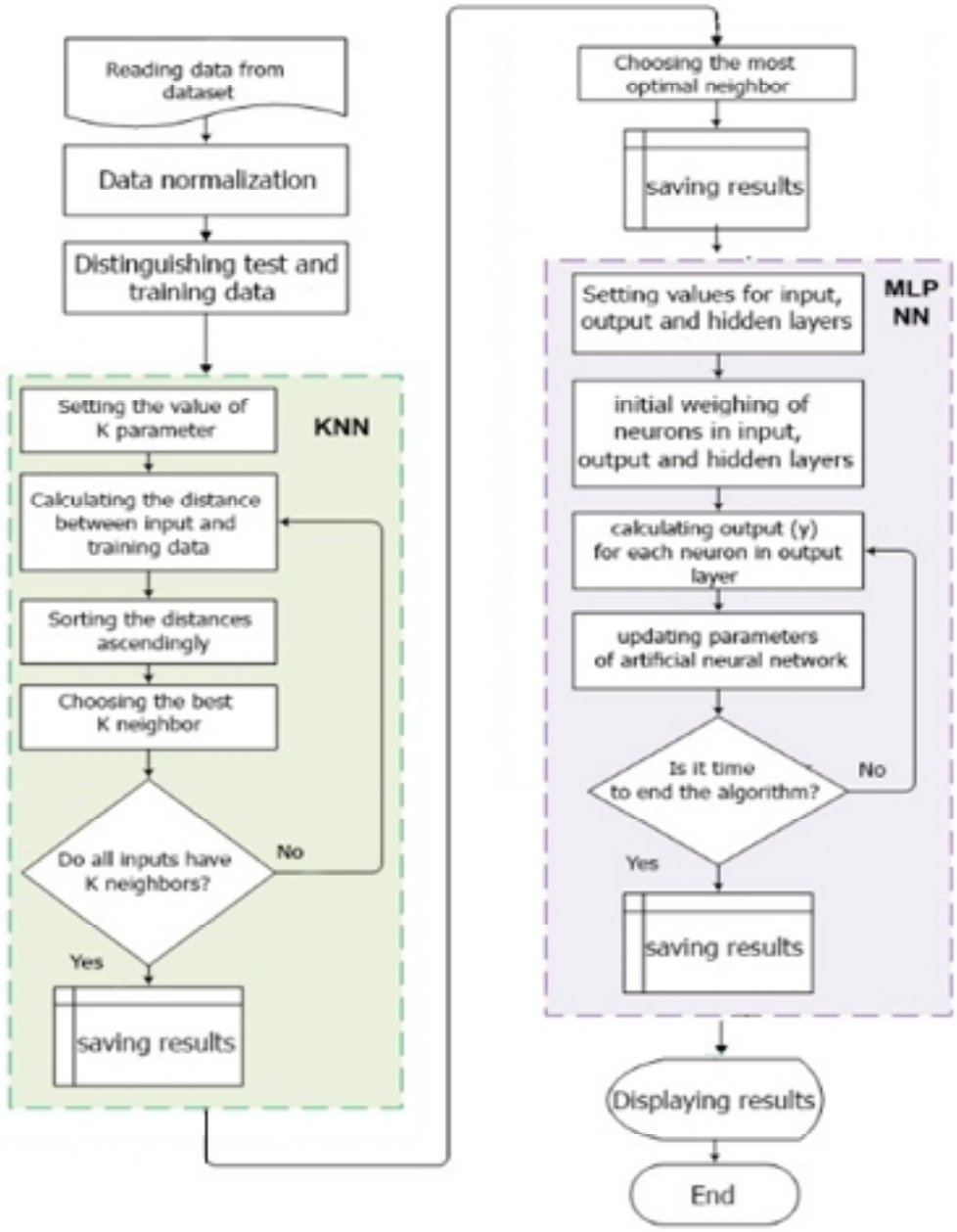

Fig. 1: The flowchart of the proposed model

In fact, ANN MLP estimates costs of projects and then redirects the projects to the KNN algorithm to re-estimate the costs. Results of the two algorithms are combined and displayed as final outcome. The algorithms are evaluated by the proposed model using Kappa coefficient as an accuracy parameter extracted from error matrix. Kappa coefficient calculates accuracy of classification in relation to a totally random classification. It is defined by Eq. (2) (Carletta, 1996).

$$\hat{K} = \frac{n\sum_{i=1}^{k} n_{ij} - \sum_{i=1}^{k} n_{i+}n_{+i}}{n^2 - \sum_{i=1}^{k} n_{i+}n_{+i}} \qquad (2)$$

Where n is total number of data, n_{i+} is the set of elements of i^{th} row and n_{+i} is the set of elements of i^{th} column. When Kappa coefficient equals zero, it is inferred that classification is done randomly without obeying any rules. Values greater than zero indicate a certain level of accuracy. The value 1 implies a completely true classification based on samples. Fig. 2 demonstrates the Pseudo code of the proposed model.

Input: all datasets (including factors affecting cost estimation per project)

Output: classified data and estimated cost per project:

Step 1: reading and normalizing existing data in datasets
Step 2: distinguishing training and test data
Step 3: retrieving KNN model
Step 3.1: setting value for parameter k
Step 3.2: estimating the distance between input data and training data
Step 3.3: sorting distances in an ascending pattern
Step 3.4: choosing the best k neighbor

Step 3.5: repeating steps 2-4 until the algorithm is over
Step 4: saving results
Step 5: choosing the most optimal neighbor
Step 6: saving results
Step 7: retrieving MLP model
Step 7.1: setting values for number of input, output and hidden layers
Step 7.2: primary weighing of existing neurons in input, output and hidden layers
Step 7.3: calculating the output (y) for each neuron in output layer
Step 7.4: updating MLA parameters
Step 7.5: repeating steps 3-4 until the algorithm is over
Step 8: saving results
Step 9: end of hybrid model
Step 10: displaying results
Step 11: end

Fig. 2: The pseudo code of the proposed model

The first step to be accomplished in the proposed model is to normalize the data, followed by classification of test and training datasets in a 20 to 80 ratio. In data classification, data are randomly selected from a dataset and no replicated data can be found in two datasets. After classification the values of ANN parameters are set and number of input, output and hidden layers are determined. Number of neurons in the input layer equals number of factors influencing cost estimation (7), while it is found to be 8 and 6 for hidden and output layers respectively. Furthermore, ANN consists of three input, output and hidden layers. Operations on training dataset start when values of ANN parameters are set, after which regulated parameters are stored in the system to be used in cost estimation of test dataset. When the KNN algorithm is completed, the most similar neighbors are built and chosen for each project to be saved. After that, neural network algorithm is started and then output of this algorithm saved and displayed as the system output.

4. Evaluation and results

In this section, datasets from projects collected by S.R.KHAZE are used to show the performance of the proposed model. Evaluation of the model is done by MATLAb 2013b.

4.1. KNN

Results of evaluating the KNN algorithm are given in Fig. 3 where true and false classification of projects by KNN is shown as a histogram chart. As can be seen, this algorithm has given only 4 false classifications of the projects.

Fig. 4 demonstrates true and false classification of projects by KNN is shown in a linear form.

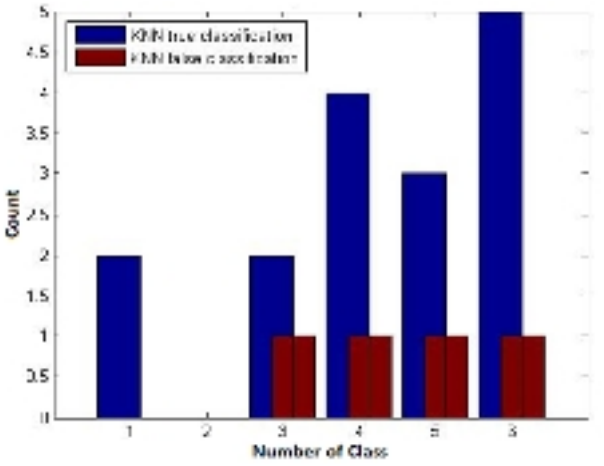

Fig. 3: The histogram of true and false classification of test dataset by KNN

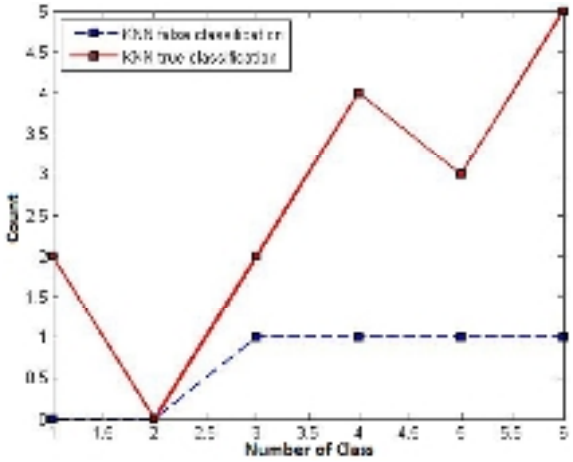

Fig. 4: Line chart of true and false classifications of test dataset by KNN

Results of evaluation by Kappa coefficient on KNN algorithm are given by Table 1. The rows in this table indicate actual values and the columns indicate predicted values. For example, in class 6 we can see 5 projects are truly classified in class 6 while 1 project is classified falsely in class 5.

Table 1: Performance of Kappa coefficient on KNN algorithm in test dataset

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Class 1	2	0	0	0	0	0
Class 2	0	0	0	0	0	0
Class 3	0	0	2	1	0	0
Class 4	0	0	1	4	0	0
Class 5	0	0	0	1	3	0
Class 6	0	0	0	0	1	5

4.2. MLP

Results of evaluating the MLP algorithm are given in Fig. 5 where true and false classification of projects by MLP is shown as a histogram chart. As

can be seen, this algorithm has given only 5 false classifications of the projects.

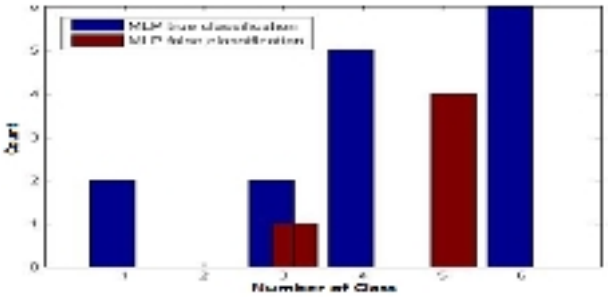

Fig. 5: The histogram of true and false classification of test dataset by MLP

Fig. 6 demonstrates true and false classification of projects by MLP is shown in a linear form.

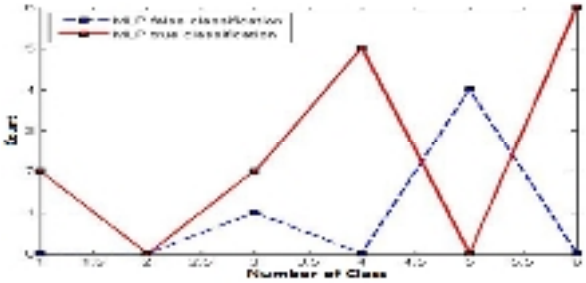

Fig. 6: Line chart of true and false classifications of test dataset by MLP

Results of evaluation by Kappa coefficient on MLP algorithm are given by Table 2. For example, in class 5 all 4 projects are falsely classified in class 4.

Table 2: Performance of Kappa coefficient on MLP algorithm in test dataset

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Class 1	2	0	0	0	0	0
Class 2	0	0	0	0	0	0
Class 3	0	1	2	1	0	0
Class 4	0	0	0	5	0	0
Class 5	0	0	0	4	0	0
Class 6	0	0	0	0	0	6

4.3. Evaluation of proposed model on training dataset

Results of evaluating the proposed model on training dataset are given in Fig. 7. Number of true and false classifications of each class by proposed model is shown in the form of a histogram chart. Fig. 7 clearly shows that all the projects of training datasets are truly classified.

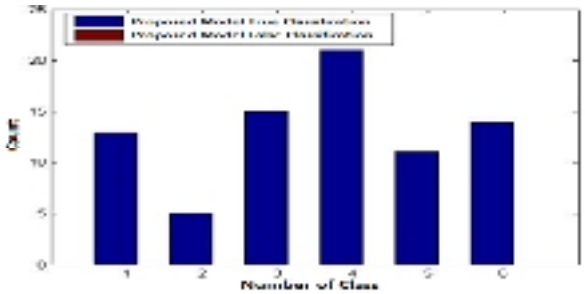

Fig. 7: The histogram of true and false classification of training dataset by the proposed model

True and false classifications of projects by the proposed model are demonstrated in a linear form in Fig. 8.

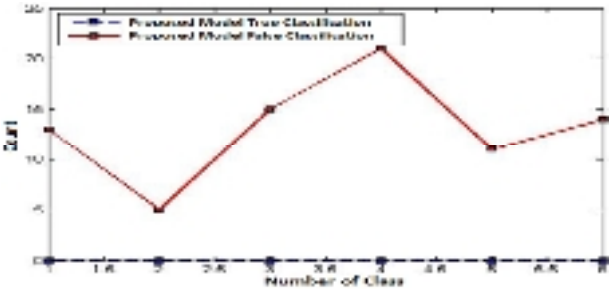

Fig. 8: Line chart depicting true and false classifications of training dataset by the proposed model

Table 3 reveals results of evaluation by Kappa coefficient on the proposed model. It shows that all the projects in the training dataset are truly classified.

Table 3: Performance of Kappa coefficient on the proposed model in training dataset

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Class 1	13	0	0	0	0	0
Class 2	0	5	0	0	0	0
Class 3	0	1	15	0	0	0
Class 4	0	0	0	21	0	0
Class 5	0	0	0	4	11	0
Class 6	0	0	0	0	0	14

4.4. Evaluation of proposed model on test dataset

Results of evaluating the proposed model on test dataset are given in Fig. 9. Number of true and false classifications of each class by proposed model is

shown in the form of a histogram chart. Fig. 9 shows that one project of test dataset is falsely classified.

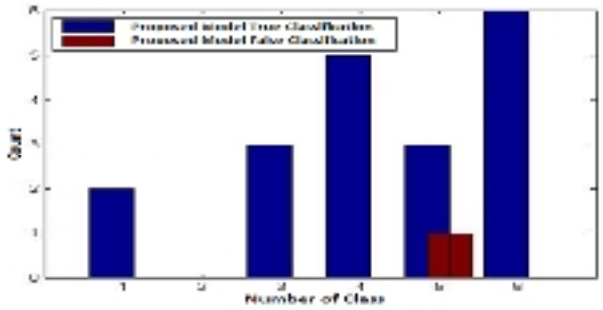

Fig. 9: The histogram of true and false classification of test dataset by the proposed model

True and false classifications of projects by the proposed model are demonstrated in a linear form in Fig. 10.

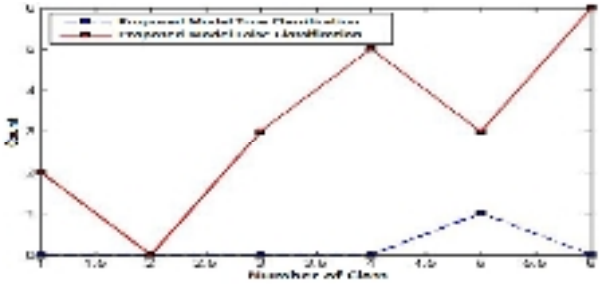

Fig. 10: Line chart depicting true and false classifications of test dataset by the proposed model

Table 4 reveals results of evaluation by Kappa coefficient on the proposed model. It shows that only one project of class 5 is falsely classifies in class 4 but all the other projects in test dataset are classified truly.

Table 4: Performance of Kappa coefficient on the proposed model in test dataset

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Class 1	2	0	0	0	0	0
Class 2	0	0	0	0	0	0
Class 3	0	1	3	0	0	0
Class 4	0	0	0	5	0	0
Class 5	0	0	0	1	3	0
Class 6	0	0	0	0	0	6

Results of evaluating KNN algorithm, MLP model and the proposed model on test datasets are given in Fig. 11 and Table 5. All the models are compared and evaluated and the results are given in Fig. 11. As can be seen, the proposed model shows a better performance than KNN algorithm, MLP model.

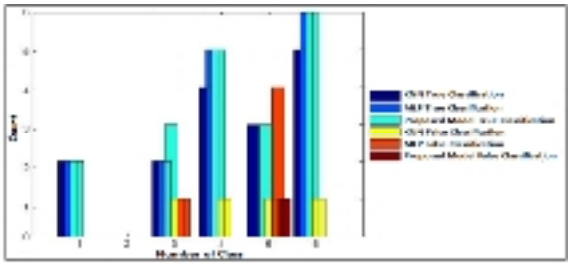

Fig. 11: Comparing histogram charts of true and false classifications on test dataset by KNN algorithm, MLP model and the proposed model

Table 5 shows results of evaluating KNN algorithm, MLP model and the proposed model on test dataset. It is clearly obvious that the percentage of true prediction and Kappa coefficient prediction for the proposed model are 0.95 and 0.93, respectively.

5. Conclusion and further works

Cost estimation of websites is an essential component for website developers if they are willing to implement successful projects. Cost estimation is usually performed in early stages of the projects by taking into account some certain project attributes and required facilities.

Table 5: Evaluation of KNN, MLP and the proposed model in test dataset

Models	percentage of true prediction	Kappa coefficient
MLP	0.75	0.67
KNN	0.80	0.74
Proposed Model	0.95	0.93

However, considering the fact that no comprehensive information is available when the project is initiated, cost estimation significance and demands new creative solutions. The present study proposes a hybrid model of KNN and ANN MLP algorithms to achieve higher accuracy and lower rate of error in cost estimation. Results of the proposed model are compared with that of KNN and ANN MLP models and imply that the proposed model yields a higher percentage of true classifications. Also, results demonstrate that Kappa coefficient and percentage of true classification 0.93 and 0.95 for the proposed model, 0.67 and 0.75 for MLP, and 0.74 and 0.80 for KNN. This paper wishes to propose a more effective model for WCS by combining data mining and machine learning in the future.

References

Carletta, Jean. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. Computational Linguistics, 22(2): 249-254.

Costagliola G., Martino S.D., Ferrucci F., Gravino C., Tortora G., Vitiello G. (2006). Effort Estimation Modeling Techniques: A Case Study for Web Applications, Proc. Sixth Int'l Conf. Web Eng., pp. 9-16.

Gharehchopogh, F.S. (2011), "Neural Networks Application in Software Cost Estimation: A Case Study", 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), pp. 69-73, IEEE, Istanbul, Turkey, 15-18.

Gharehchopogh, F.S., Khaze, S.R, Makeli, I., (2015), "A New Approach in Bloggers Classification with Hybrid of K-Nearest Neighbor and Artificial Neural Network Algorithms ", Indian Journal of Science and Technology, Vol: 8, No: 3, pp: 237–246.

Khaze S.R., Ghaffari A., Masdari M. (2013). Using the Naïve Bayes Algorithm for web Design Cost Estimation with Content Management System, International Journal of Advanced Research in Computer Science and Software Engineering, 3 (11): 999-1007.

Martin, B. (1995) Instance-Based Learning: Nearest Neighbour with Generalisation, Doctoral dissertation, University of Waikato.

Mendes E. (2014). Introduction to Web Resource Estimation, Practitioner's Knowledge Representation, pp. 55-60.

Mendes E., Martino SD., Ferrucci F., Gravino C. (2007). A Replicated Study Comparing Web Effort Estimation Techniques, Web Information Systems Engineering (WISE 2007), Lecture Notes in Computer Science, Vol. 4831, pp. 423-435.

Mendes, E., Mosley, N., & Counsell, S. (2001). Web Metrics - Estimating Design and Authoring Effort. IEEE Multimedia, Special Issue on Web Engineering, pp. 50-57.

Mendes, E., Mosley, N., & Counsell, S. (2002), The Application of Case-based reasoning to Early Web Project Cost Estimation, Proceedings COMPSAC 02, pp. 173-183

Mendes, E., Watson, I., Triggs, C., Mosley, N., & Counsell, S. (2002b). A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications. Proceedings Metrics 2002, pp. 131-140.

Mo Q., Xiao F., Su D.Z. (2014). Design and Implementation of Enterprise Resources Content Management System, Knowledge Engineering and Management, Advances in Intelligent Systems and Computing, Vol. 278, pp. 259-268.

Reifer, D. J (2000). Web development: Estimating Quick-to-market Software. IEEE Software, pp. 57-64

Ruhe, M. Jeffery, R. & Wieczorek, I. (2003), Cost estimation for Web applications, Proceedings ICSE 2003, pp. 285-294.

S. German, E. Bienenstock, R. Doursat. (1992). Neural Networks and the Bias/Variance Dilemma, Neural Computation, 4(1): 1-58.

Seel N.M. (2012). Learning Content Management System, Encyclopedia of the Sciences of Learning, Springer US.