REVIEW ON SOFTWARE EFFORT ESTIMATION BY DATAMINING AND MACHINE LEARNING APPROACHES

Rupsi Bedi[1], Monika Pathania [2]
[1, 2] *Computer Science, Bells Institute of Management and Technology, Shimla, INDIA*

***Abstract*- In the field of software development, software project estimation is the most challenging task. If there is no proper and reliable estimation provided in the software development, there will be no proper arrangement as well as control of the project. Software project estimation is necessary to handle underestimates and overestimates in terms of cost, effort etc. Small projects are not difficult to estimate and accuracy can be improved by traditional approach of Expert judgment. As the measure of project size increases i.e. for embedded and large-scale projects, precision and accuracy become important concern. In this thesis feature set is increased by adding Line of code (LOC). Effort estimation process starts after the estimation of size of the project. This estimation performed after the complete requirements are defined and size mentioned.**

Keywords-. Software project estimation, Line of code.

I. INTRODUCTION

In the area of software development, software project estimation is the most challenging task. If there is no proper and reliable estimation provided in the software development, there will be no proper arrangement as well as control of the project. Even when all the important factors are taken into consideration during the software development process still projects are not accurately estimated. It doesn't utilize estimates for improving the development of software. When a project is underestimated the effects such as under-staffing, under-scoping the quality assurance effort and missing the deadlines resulting in loss of credibility are seen [18]. Software project estimation is necessary to handle underestimates and overestimates in terms of cost, effort etc. [1]. If a project is given more number of resources than it actually requires the resources will be utilized by it but the cost of the product increases, due to this reason deployment of estimation techniques is essential.

Small projects are not difficult to estimate and accuracy can be improved by traditional approach of Expert judgment. As the measure of project size increases i.e. for embedded and large-scale projects, precision and accuracy become important concern. Estimating the effort with a huge value of reliability is an issue which has not been unraveled yet.

1.1 Software Project Estimation

Software Project Estimation is essential as already stated from the literature. For the purpose of project estimation, these four steps are considered [3] [2]:

1) Development product's size estimation: There are Lines of Code (LOC) as well as Function Points (FP) which help in this estimation. However, various other methods are also used for measuring the estimation such as Use case points (UCP), Story points etc. There are certain merits as well as demerits of this estimation.

2) Effort estimation in terms of person-month or person-hours.

3) Scheduling estimation for months of a calendar.

4) Project cost estimation in dollars or any other local currency.

1.1.1 Estimating size

The first step towards achieving an effective project estimate is to provide an accurate size estimate of the software. Along with the formal descriptions of the requirements for project, scope of size estimation might start [4]. Various examples of the data present for cost estimations are requirement specifications of customer, proposal requests and the specification of a system or software requirement. There can be additional details provided with the help design documents for the chances that a re-estimation can be performed by the project at its later phases of the lifecycle [21].

For the purpose of product size estimation, the two ways are described below:

1) By analogy: For the purpose of creating a new project, it is easy to provide all the required estimations if one have already dealt with the similar kind of projects earlier. It is much likely similar to the previous one which helps in providing the total cost of the project according to its size. The total size estimates to the percentage of a small piece of previous project. When estimating the size of new project, the sizes of all smaller pieces are to be included. With the help of analogy, an experienced estimator can create better size estimates. Only when there are accurate size values of the previous project available and the new project is similar to the previous one, only then can this technique be efficient [7].

2) Counting product features along with an algorithmic approach: Function Points are for instance used as an algorithmic approach to convert the tally into an estimate of size. There are numerous subsystems, classes/modules,

methods/functions, which are included in the macro-level "product features".

1.1.2 Estimating effort

After receiving the size estimation of a product, it becomes very easy to estimate the effort of it. When the software development lifecycle of a project is defined only then the conversion from software size to total project effort estimation is possible. Further the designing, develop and test of the software are defined for project development. In addition to coding of the software there is much more to the software development project [11]. The smallest part of the effort is basically the coding part. The larger part here includes the writing and reviewing documents, implementation of the prototypes, deliverable designing, reviewing as well as testing of the code include the larger part of the project effort. There are certain guidelines provided for the purpose of identifying, estimating as well as summing up each of the activity performed for constructing the product of certain size.

For the purpose of deriving effort from size, there are two main ways:

1. The historical data of the organization itself is the most helpful thing which helps in providing estimates of which project had what size and costs with respective to each other [12]. There are certain factors included here which are:

(a) Documentation of actual results using previous projects.

(b) There should be minimum one project in the past which has similar size to that of the one being newly proposed. This helps in determining estimations of projects which would cost the same with similar size.

(c) A similar development lifecycle is to be developed for developing methodology which uses similar tools, team with same skills and experience which will help the new project.

2. The most accepted and appreciated approach can be selected for the cases where there are no projects for providing help in estimating the cost and size of the projects. This case can occur if you have not gathered any of your previous projects or when the project you are developing is very new and not similar to any of the earlier projects. Barry Boehm's COCOMO or the Putnam Methodology is example of the effort models used for estimation. They are used to convert size estimate into effort estimate [13] [15]. They are valuable but are not that accurate as that of the own historical data of an organization and the accuracy varies as per different scenarios and application areas.

1.1.3 Estimating schedule

The third important step in software development project is the determination of project schedule from the effort estimate. The number of people working on a project, the type of work they will do, the starting time and ending time of the project are the factors that are to be involved here. The data gathered from this step is to be laid on to the calendar schedule. This also helps in determining the way in which the work can be broken down into a proper schedule [14].

1.1.4 Estimating Cost

Total cost of the project can be estimated through various components such as labor, hardware and software or rentals, travel for meeting or testing related issues, telecommunications, instructional classes, office area, etc. All of the enlisted factors affect the estimation of effort in one way or the other.

The total estimate of the project is also dependent in a way on the amount of cost an organization allocates to it. In some of the fields the cost allocation is not done at all and its adjustment is done by increasing the labor costs per hour. The overall costs for the software development are estimated by the software development project manager accordingly [6]. To determine what percentage of total project effort ought to be allocated to every position, historical data or industry data models offer assistance.

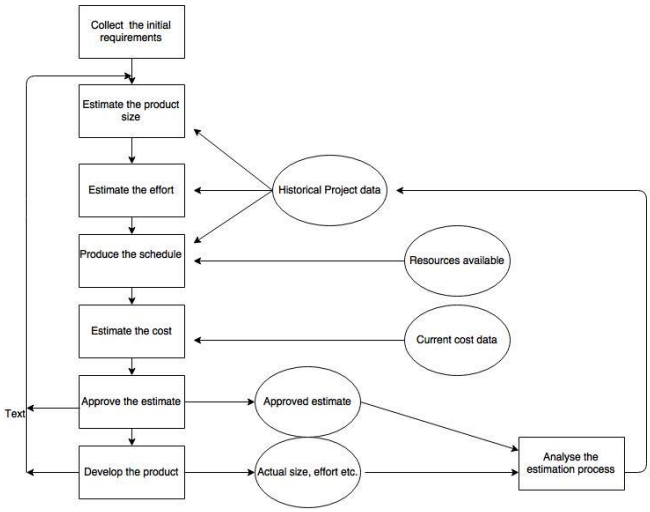

Figure 1: Software Project Estimation process

1.2 Models for Software Effort Estimation

Software cost estimation model is a backhanded measure, which is utilized by software personnel to predict the cost of a project. The development of software product shifts depending upon the earth in which it is being developed. For projects with familiar environment it is anything but difficult to predict the cost of the project [4] [6].

For the organization to develop a cost estimation model the following things are required.

- List important or critical cost drivers.
- Prepare a scaling model for every cost driver.
- Find projects with similar environments.
- Compare the project with previous familiar projects.
- Evaluate the project whether it is feasible inside the budget constraints.
- Incorporate the critical features in an iterative manner.

Cost drivers are those critical features which affect the project. The cost drivers may vary the cost of building a project. The most important cost driver is size of the project. Size of the

project is measured in Kilo lines of code (KLOC). Function points are the empirical measurement to measure size of the project.

1.2.1 Constructive Cost Estimation Model (COCOMO)

COCOMO model was proposed by Barry Boehm in 1981 and is a widely adopted and accepted algorithmic cost and effort estimation model. It makes use of parameters and evaluation equation from historical software project experiences for estimation process. It employs metrics for measurement of effort and effort and scale parameters for calculating effort requisite for the project [6] [18]. Although, COCOMO model is used largely but it has limitations for deployment in today's software development process.

1.2.2 Existing Estimation Models

Table 1. Advantages and Disadvantages of Existing Methods.

Method	Type	Advantages	Disadvantages
COCOMO	Algorithmic	Clear results, very common	Much data is required, It is not suitable for any project,
Expert Judgment	Non-Algorithmic	Fast prediction, Adapt to especial projects	Its success depend on expert, Usually is done incomplete
Function Point	Algorithmic	Language free, Its results are better than SLOC	Mechanization is hard to do , quality of output are not considered
Analogy	Non-Algorithmic	Works based on actual experiences, having especial expert is not important	A lots of information about past projects is required, In some situations there are no similar project
Parkinson	Non-Algorithmic	Correlates with some experience	Reinforces poor practice
Price to win	Non-Algorithmic	Often gets the contract	Generally produces large overruns
Top-down	Non-Algorithmic	Requires minimal project detail, Usually faster and easier to implement, System level focus	Less detailed basis , Less stable
Bottom-up	Non-Algorithmic	More detailed basis, More stable , encourage individual commitment	May overlook system level costs , Requires more effort, More time consuming

1.3 Proposed Algorithm

1.3.1 Water cycle Optimization: It is a population based meta-heuristics algorithm which is used to simulate the leadership hierarchy and hunting mechanism of grey wolves in nature. The main concept of water cycle optimization algorithm is to simulate the behavior of water cycle which are living in a pack. We use to assume fitness solution as alpha, beta and delta. Alpha is known as the level leaders and is responsible for decision making in the pack. The wolf pack persistence is based on the decision of alpha. Beta is known as the second level subordinate wolves. The beta operation is for help in making the decision for alpha or other activities. Delta is known as the third level subordinate wolves. This category member consists of elders, scouts, hunters, caretakers, and sentinels. For region boundary observation and in any danger case, scouts are liable for the warning. The protection and pack's safety guarantee is given by sentinels [18]. The expertise wolves are the elders, denoted as alpha or beta. Alphas and betas are helped by hunters while prey hunting and caring for the ill, weak, and wounded wolves by caretakers and providing food for a pack. Omega is the lowest level. All dominant wolves with omega wolves have to comply. In the mathematical model hunting behavior of grey wolves has been assumed as alpha, beta, and delta have good knowledge about the potential location of prey. Grey wolves have the ability of memorizing the prey position and encircling them. The alpha as a leader performs in the hunt. For simulating the grey wolves hunting behavior in the mathematical model, assuming the alpha (α) is the best solution. The second optimal solution is beta (β) and the third optimal solution is delta (δ). Omega (ω) is assumed to be the candidate solutions. Alpha, beta, and delta guide the hunting while position should be updated by the omega wolves by these three best solutions consideration [18].

Application of Water cycle Optimization

- It gives quality result for fuzzy portfolio selection.
- They are used to solve the economic dispatch problems.
- It can be used to select the feature set for feature selection.
- Exploration and exploitation is done properly from adaptive values by using GWO algorithms.

II. RELATED WORK

Several researches have been done in the recent years. A Literature Survey has been done from year 2009 to 2017. The Literature Survey is presented below in the form of four

sections. The first section consists of the studies undertaken to study Artificial Neural Networks (ANN) based techniques for effort estimation. The second section elaborates work done based on Fuzzy Logic .The third studies the optimization algorithms and other estimation techniques.

2.1 Artificial Neural Networks (ANN) Based Estimation

Ali Bou Nassif et al. [3] in 2013 presented Log Linear Regression (LLP) and Multi-Layer Perceptron Models (MLP) for effort estimation of software based on Use case points. The result of the two models shows their performance was better for small projects than for large sized projects. Ali Bou Nassif et al. [2] in 2012 proposed a Treeboost Model based on stochastic gradient boosting which employed Use case points (UCP). Software size in the form of UCP's, complexity and productivity were inputted to the Treeboost model and the results were evaluated against the Multiple Linear Regression model. The Treeboost Model results were promising. The performance metrics used are Mean magnitude relative error (MMRE), Prediction (PRED), Mean Square Error (MSE) and Median of Magnitude of Relative Error (MdMRE). Shashank Mouli Satapathy et al. [4] in 2014 proposed a machine learning techniques based model of Stochastic Gradient Boosting (SGB) to improve the software effort estimates using class point approach. The effort parameters are improved by using SGB model resulting in improved prediction accuracy. A comparative analysis of SGB along with Multi-Layer Perceptron and the Radial Basis Function Network has been presented. Shashank Mouli Satapathy et al. [9] in 2016, proposed an approach which is based on use case points (UCP) and random forest classifier for effort estimation. In the proposed scheme, the UCP approach has estimates the UCP's considering the size, complexity, productivity and also the actual effort values of 149 projects. The random forest classifier has classified impact of each of factors. The random forest classifies features by building decision trees and it is implemented by considering various case studies. The simulation results shows that proposed approach performs well in terms of Mean magnitude relative error (MMRE) and Prediction (PRED) performance metrics.

Poonam Rijwani and Sonal Jain [10] in 2016, presented a Multilayer Feed Forward (MLFFN) Artificial Neural Network Technique for enhanced effort estimation. COCOMO 81 dataset was employed consisting of 63 software projects. The validation method used was EpochesUntil tolerance level>.999. The results provided reduced MRE in MLFFN than in COCOMO.

3.2 Fuzzy Logic Based Estimation

Anish Mittal et al. [1] in 2010, proposed an Enhanced Fuzzy system for enhancing estimations of COCOMO model by incorporating Triangular Fuzzy function. The results were evaluated for a firm dataset and were promising. Monika and Om Prakash Sangwan [17] in 2017 presented an analysis of various machine learning techniques used in software effort estimation. The machine learning techniques employed so far are based on Artificial Neural Network, Fuzzy Logic, Analogy Based Estimations, Genetic Algorithm and other techniques. The paper highlights relevance of each techniques depending upon its own nature and environment in which it employed.

3.3 Others Techniques and Optimization algorithms

David L. Gonzalez-Alvarez, et al. [5] in 2014 proposed that proteins are molecules that shape the mass of living creatures. The proteins exist in dissociated shapes like amino-acids. These proteins help in completing different biological function in living beings and there are no such functions that could be performed without them. Due to this reason the analysis of proteins has become a very important issue in biology. The identification of conserved patterns in an arrangement of related protein sequences can help in achieving biological data about these protein functions. This paper displayed a novel algorithm based on Teaching Learning Based Optimization (TLBO) which is combined with a local search work. This work is specialized to predict common patterns in sets of protein sequences [11]. These were used to evaluate the performance of the proposed technique. The proposed technique helped in making predictions and also helped in improvement of quality of the solutions which were found by biological tools. Chalotra et al. [9] in 2015 introduced that the target of momentum examination was applying Bee Colony Optimization (BCO) meta-heuristic way to deal with enhance the parameters of COCOMO model for enhancing software cost estimation. BCO methodology a "bottom-up" way to deal with modeling where uncommon sorts of artificial agents are made by similarity with bees which are utilized to take care of complex combinatorial optimization issues. The proposed model validation was done utilizing Interactive Voice Response software venture dataset of an organization [10]. The BCO approach creates different partial arrangements and best arrangement is chosen based on Mean Magnitude of Relative Error. The results acquired demonstrate that the proposed BCO based model can enhance the precision of cost estimation furthermore beat different models. S. M. Sabbagh Jafari, F. Ziaaddini [11] in 2016, presented a meta-heuristic optimization algorithm for predicting effort estimation for developing project. The paper presents an approach to optimize the Constructive Cost Estimation Model (COCOMO) using Meta-heuristic Harmony Search Algorithm. The NASA dataset was used to evaluate the model. The Proposed model optimized the Mean Magnitude Relative Error (MMRE) to nearly 21%. Peyman Khazaei, et al. [13] in 2016 proposed that in the day-ahead power systems scheduling, system operators formulated and solved the unit commitment (UC) problem to determine ON/OFF status and power dispatch of the producing units. Although different methods had been exhibited to solve the UC problem, it was a blended integer optimization problem which was elusive its global optimum solution. In the paper, the teaching-learning-based optimization (TLBO) technique, which as an

evolutionary algorithm, was employed to solve the unit commitment problem. The proposed algorithm was compared with several existing methods. The numerical results showed the effectiveness of the proposed TLBO which was compared with several well -known evolutionary algorithms, i.e., DP, PSO, and SLF A. The TLBO not just gave a solution bring down operating costs, additionally had a lower computation time. In addition, adequate spinning reserve was given to alleviate the effect of rapid load/generation changes because of unexpected disturbances. Yu-Huei Cheng [12] in 2016 proposed that numerous single nucleotide polymorphisms (SNPs) for complex genetic diseases were genotyped by polymerase chain response restriction piece length polymorphism (PCR-RFLP). A feasible PCR-RFLP primer match was to be designed. Also there was a need to discover accessible restriction enzymes which could perceive the target SNP for PCR experimental purposes. It was developed to improve mutagenic primer and it used the idea of "teaching-learning for searching more
feasible mutagenic primers. It gave the latest accessible restriction enzymes .The proposed method included more accurate formulas of GC-based melting temperature and thermodynamic melting temperature. These were used for the calculation of melting temperature. Mutagenic grid was reserved to build the efficiency of judging a hypothetical mutagenic primer if involved accessible restriction enzymes for perceiving the target SNP. Dragicevic Srdjana, et al [15] in 2017 proposed a Bayesian network model for predicting software effort estimation for agile software development .The proposed model predicts effort irrespective of the agile methods available and the input data is elicited easily. The precision and prediction accuracy achieved by the proposed Bayesian network are very favourable. The above sections presented literature review of Software effort estimation techniques that have been employed using machine learning methods and of optimization methods. The machine learning algorithms have outperformed and provide better accuracy due to their learning natures. ANN has been used dominatingly in effort estimation models but have complex algorithms and need to be optimized.

2.4 Water cycle Optimization*:*

Mohanty, et al. [14] in 2016 proposed a design termed as maximum power point tracking (MPPT) for the photovoltaic system by using water cycle technique. GWO is a technique used to beat the limitation of lower tracking efficiency, steady-state oscillations, and transients, as encountered in, perturb and observe (P&O) and improved PSO (IPSO) techniques. By using GWO-based MPPT author used to solve the issues during tracking the global peak of the photovoltaic array under partial shading conditions. This proposed algorithm is used to implement the photovoltaic system by using mat-lab and SIMULINK. The experimental and simulated result shows that proposed algorithm gives better performance in both P&O and IPSO MPPTs.

Table.1 Existing Scheduling Model

Author's Name	Year	Methodology Used	Proposed Work
Shashank Mouli Satapathy *et al.*	2014	Gradient Boosting (SGB)	Proposed a machine learning techniques based model of Stochastic Gradient Boosting (SGB) to improve the software effort estimates using class point approach.
Anish Mittal *et al*.	2010	COCOMO model	Proposed an Enhanced Fuzzy system for enhancing estimations of COCOMO model by incorporating Triangular Fuzzy function.
David L. Gonzalez-Alvarez, *et al.*	2015	Teaching Learning Based Optimization (TLBO)	Proposed that proteins are molecules that shape the mass of living creatures. The proteins exist in dissociated shapes like amino-acids.
Yu-Huei Cheng	2016	PCR-RFLP	Proposed that numerous single nucleotide polymorphisms (SNPs) for complex genetic diseases were genotyped by polymerase chain response restriction piece length polymorphism (PCR-RFLP).
Sulaiman, *et al.*	2010	Optimal Reactive Power Dispatch	Presents a new meta-heuristic technique named as Water cycle Optimizer (GWO) which is influenced from leadership of grey wolves and their hunting

			behavior to solve problems in the optimal reactive power dispatch (ORPD).

Mirjalili, et al. [16] in 2016 proposes a multi-objective water cycle optimizer (MOGWO) in the purpose of optimizing problems or issues in multiple objectives for the first time. In this method constant size external archive is developed to the GWO to save and retrieve the Pareto optimal solutions. Then this archive is used to employ to define the social hierarchy and simulate the hunting behavior of grey wolves in multi-objectives search spaces. This proposed algorithm is tested on 10 multi-objective benchmark problems and used to do a comparison with two known metaheauristics 1) Multi-objective Evolutionary Algorithm 2) Multi objective PSO. The comparative result shows that this proposed algorithm is capable to give competitive result and performs quite better than other algorithm. Sulaiman, et al. [7] in 2015 author presents a new meta-heuristic technique named as Water cycle Optimizer (GWO) which is influenced from leadership of grey wolves and their hunting behavior to solve problems in the optimal reactive power dispatch (ORPD). This is a problem like major problem in power system where author applies GWO to resolve the problems. GWO is used to determine appropriate combination of control variable like generator voltages, tap changing transformers ratio etc. In this paper, to demonstrate the effectiveness of GWO technique by using two case studies IEEE 30-bus system and IEEE 118-bus system . The result demonstrates that GWO can attain minimum power loss and voltage deviation. Zhu, et al. [8] in 2015 author used to introduce a new meta-heuristic method to improve the present meta-heuristic method by using hybrid water cycle optimization for global optimization and test scheduling for 3-D stacked system-on-chip. This proposed algorithm integrates the convergence speed of GWO and improves its performance. To identify the performance of the proposed algorithm 23 known benchmark function and NP hard problems of test scheduling for 3-D is considered. The experimental result demonstrates that the proposed algorithm gives better performance to manipulate the optimum and beneficial for explorations. Pradhan, et al. [15] in 2016 authors present a new evolutionary optimization approach termed as water cycle optimization on the basis of grey wolves, for the optimal operations of economic load dispatch (ELD). The concept of GWO which is applied to non-linear ELD problems is reliable. This algorithm is applied and test on 4 test system having 10, 40, 80 and 140 units. The result demonstrates that the proposed algorithm has some significance to the proposed algorithm as compare to others. GWO is very effective to solve ELD problems.

III. CONCLUSION

Software effort estimation is a challenging issue in the software development process. There are various methods that are proposed by the researchers to solve this issue. In this thesis accuracy of the prediction is improved by feature selection and Machine Learning approach. In this work features selection approach is done by using Grey wolf optimization algorithm. GWO algorithm is used to select the effective weighted feature. The result is shown by the analysis process

IV. REFERENCES

1] Mittal, A., Parkash, K., & Mittal, H. (2010). Software cost estimation using fuzzy logic. *ACM SIGSOFT Software Engineering Notes*, *35*(1), 1-7.

[2] Nassif, A. B., Capretz, L. F., Ho, D., & Azzeh, M. (2012, December). A treeboost model for software effort estimation based on use case points. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on* (Vol. 2, pp. 314-319). IEEE.

[3] Nassif, A. B., Ho, D., & Capretz, L. F. (2013). Towards an early software estimation using log-linear regression and a multilayer perceptron model. *Journal of Systems and Software*, *86*(1), 144-160.

[4] Satapathy, S. M., Acharya, B. P., & Rath, S. K. (2014). Class point approach for software effort estimation using stochastic gradient boosting technique. *ACM SIGSOFT Software Engineering Notes*, *39*(3), 1-6.

[5] Gonzalez-Alvarez, D. L., Vega-Rodriguez, M. A., & Rubio-Largo, A. (2015). Finding patterns in protein sequences by using a hybrid multiobjective teaching learning based optimization algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, *12*(3), 656-666.

[6] Chalotra, S., Sehra, S. K., Brar, Y. S., & Kaur, N. (2015). Tuning of COCOMO model parameters by using bee colony optimization. *Indian Journal of Science and Technology*, *8*(14), (pp.2-8)

[7] Sulaiman, Mohd Herwan, et al. "Using the gray wolf optimizer for solving optimal reactive power dispatch problem." *Applied Soft Computing* 32 (2015): 286-292.

[8] Zhu, Aijun, et al. "Hybridizing water cycle optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC." *Journal of Systems Engineering and Electronics* 26.2 (2015): 317-328.

[9] Satapathy, S. M., Acharya, B. P., & Rath, S. K. (2016). Early stage software effort estimation using random forest technique based on use case points. *IET Software*, *10*(1), 10-17.

[10] P. Rijwani and S. Jain. (2016) "Enhanced Software effort estimation using Multi-Layer Feed Forward Artificial Neural

Network Technique," Procedia Computer Science, vol 89, pp. 307-312.

[11] Jafari, S. S., & Ziaaddini, F. (2016, March). Optimization of software cost estimation using harmony search algorithm. In *Swarm Intelligence and Evolutionary Computation (CSIEC), 2016 1st Conference on* (pp. 131-135). IEEE.

[12] Cheng, Y. H. (2016). A novel teaching-learning-based optimization for improved mutagenic primer design in mismatch PCR-RFLP SNP genotyping. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, *13*(1), 86-98.

[13] Khazaei, P., Dabbaghjamanesh, M., Kalantarzadeh, A., & Mousavi, H. (2016, July). Applying the modified TLBO algorithm to solve the unit commitment problem. In *World Automation Congress (WAC), 2016* (pp. 1-6). IEEE.

[14] Mohanty, Satyajit, Bidyadhar Subudhi, and Pravat Kumar Ray. "A new MPPT design using water cycle optimization technique for photovoltaic system under partial shading conditions." *IEEE Transactions on Sustainable Energy* 7.1 (2016): 181-188.

[15] Pradhan, Moumita, Provas Kumar Roy, and Tandra Pal. "Water cycle optimization applied to economic load dispatch problems." *International Journal of Electrical Power & Energy Systems* 83 (2016): 325-334.

[16] Mirjalili, Seyedali, et al. "Multi-objective water cycle optimizer: a novel algorithm for multi-criterion optimization." *Expert Systems with Applications* 47 (2016): 106-119.

[17] Sangwan, O. P. (2017, January). Software effort estimation using machine learning techniques. In *Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on* (pp. 92-98). IEEE.

[18] Dragicevic, S., Celar, S., & Turic, M. (2017). Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software*, *127*, 109-119.