# Estimating Effort in Comprehensive Software Development Using NN, ABE, GAFLANN

[1]J. Sruthi, [2]Dr. Jose Moses, [3]M. Krishna Kishore
[1,2,3]Dept. of Computer Science Engineering, Raghu Engineering College, Visakhapatnam, AP, India

**Abstract**
This research will observe the use of Artificial Neural Networks for estimating software cost. Reliable effort estimation remains a progressing test to software engineers. Precise estimation is an intricate procedure since it can be envisioned as software effort expectation, as the term demonstrates forecast never turns into a real. The objective of this paper is to concentrate on the observational software effort estimation. The essential conclusion is that no single system is best for all circumstances, and that a watchful examination of the consequences of a few methodologies is well on the way to create sensible appraisals.This paper gives a similar examination of different accessible software effort estimation techniques. The research tries to compare different type of neural network architectures. The proposed ABE model increases the accuracy and each technique have different results which will be shown using matlab. These strategies can be broadly utilized as a part of this paper, a standout amongst the most prevalent effort estimation techniques is similarity based estimation (ABE), Functional Link Artificial Neural Network (FLANN), Genetic Algorithm for Optimizing Functional Link Artificial Neural Network (GAFLANN).

**Keywords**
Neural Network, FLANN, GAFLANN, Software Engineering, Effort Estimation

## I. Introduction

Software Measurement is the procedure which helps the associations to enhance an estimation program. It doesn't give direction to utilization of some estimation, for example, software cost estimation or to dissect software unpredictability. Rather than it helps on powerful software estimation program and for seeing a portion of the key lessons that how the work. Klayman et al. (1999) reported that the estimation is exact judgment process which relies on certainty individuals however arrogance increments with the trouble of the errand [1]. The reason for the Software Measurement has grown up of fruitful estimation applications. Stone and Opel (2000) report about likelihood judgment exactness by inspecting the impacts of two diverse preparing methods, for example, execution criticism or natural input. They quantified their change in adjustment and segregation as an element of criticism sort which require separate preparing systems for development [2]. It exhibits the particular techniques and exercises which assume the parts to accomplish the objectives furthermore parts of the general population included. Estimation is the procedure by which numbers or images are alloted to properties of elements in this present reality so as to portray the characteristics by unmistakably characterized rules.

In this way, estimation requires Entities (objects of interest). Qualities (attributes of substances). Standards (and scales) for appointing qualities to the properties. The most famous non-algorithmic estimation strategy is ABE technique, which was displayed in 1997 [5]. This strategy utilizes examination of a task with other comparative authentic cases. The correlation depends on the components of two tasks. Besides, other keen techniques, for example, neural system, fluffy principles and diverse strategies for information mining have been utilized as a part of effort estimation region [2, 6, 8]. Software managements as another idea should be more examined in various PC regions, for example, distributed computing and web2. Assortment and the absence of enough gauges in managements have made the important effort assessment less saw for advancement [7]. Sadly, the expression "management" has diverse implications, yet as indicated by the definition, the importance of software management is software that is exhibited by program [1–2]. Distributed computing gave another kind of management deal called SaaS (software as anmanagement) in which the implications of software and management are connected together [2]. Distinctive sorts of management arranged engineering (SOA) ventures incorporate management mining, management advancement, application improvement, management coordination, management framework, management, and management design [2]; the center of this paper is just on the second part, software managements improvement, and there is no attention on reactions, for example, equipment, base, stage, and working framework. n this paper, we are worried with cost estimation models that depend on Functional Link artificial neural systems .The Functional Link artificial neural system (FLANN) design for anticipating software improvement effort is a singlelayer sustain forward neural system comprising of one info layer and a yield layer. The FLANN creates yield (effort) by extending the underlying inputs (cost drivers) and after that handling to the last yield layer. Every info neuron compares to a part of an information vector. The yield layer comprises of one yield neuron that processes the software advancement effort as a direct weighted total of the yields of the info Layer [9]. The substantial and non-typical information sets dependably lead FLANN strategies to low forecast precision and high computational multifaceted nature. To ease these disadvantages our proposed thought has been given to at the same time upgrade chose class of activities and their component determination by Genetic Algorithm (GA).

The most common approaches for effort estimation are expert judgment, algorithmic models and analogy. Expert judgment is widely used for small projects, and sometime more than one expert's opinion is pooled for estimation. Algorithmic models such as COCOMO [10], SLIM [11], function points [12] are popular in the literature. Where $\alpha$ is a productivity coefficient and $\beta$ refers to the economies of scale coefficient. The size is measured in estimated Line Of Code (LOC). In analogy estimation, similar completed projects with known effort value are used for predicting the effort for the project.

In this paper, the principle center is not just examining the exactness of the forecast utilizing FLANN arrange additionally diminishing the computational many-sided quality of the system. The point of this study is to check if FLANN system can be utilized for expectation of effort on the premise of effort multipliers, size of the undertaking, scale component utilized as a part of task improvement. To exactly assess the preparation algorithms and to discover which preparing algorithm is reasonable for the estimation reason.

## II. Related Work

Exact and steady expectation of effort is an exceptionally urgent for the software ventures. Numerous specialists utilized their distinctive ANN and diverse dataset, to anticipate the effort all the more accurately. [1] Hareton Leung, Zhang Fan,et al.[8] gives a general review of software cost estimation strategies incorporating the late advances in the field. As some of these models depend on a software size evaluation as info, an outline of basic size measurements is given first. At that point the cost estimation models that have been proposed and utilized effectively are highlighted. This paper incorporates two noteworthy classes of models utilized as a part of cost estimation. These classes are: algorithmic and non-algorithmic. Algorithmic Methods incorporates: Functional Link Artificial Neural Network (FLANN), Genetic Algorithm for Optimizing Functional Link Artificial Neural Network (GAFLANN). Furthermore, Non-algorithmic Methods incorporates: (ABE). Each has its own qualities and shortcomings. A key element in selecting a cost estimation model is the exactness of its evaluations. Tragically, regardless of the substantial group of involvement with estimation models, the exactness of these models is not palatable. The paper incorporates remark on the execution of the current estimation models and depiction of a few more up to date ways to deal with cost estimation and its better to think about the more new methods to enhance the exactness [7]. Christos Stergiou and DimitriosSiganos, et al.[10] worries that Neural Network (NN) is a data handling worldview that is roused by the way natural sensory systems, for example, the cerebrum, process data. The key component of neural systems is its structure of the data handling framework which is made out of an expansive number of exceedingly interconnected preparing components (neurons) to take care of particular issues. Neural Networks takes care of the issue through the learning procedure. Learning in Neural Networks includes changes in accordance with the weights existing on associations between the neurons. It likewise incorporates the engineering of neural systems that is food forward systems which handle just in one course, input systems which process in both bearings. This paper likewise portray the diverse systems layers which are info layer which takes information and like the dendrites in the natural framework, next is the shrouded layer which prepare the data and works like the neural connections in organic framework last is yield layer which takes yield from concealed layer which is like the axon in the natural framework. Distinctive learning strategies like administered and unsupervised is examined with exchange capacity. The conduct of neural systems relies on the weights and the exchange capacity. Neural systems are best at recognizing examples or patterns in information, they are appropriate for expectation or determining needs.

## III. Cost Estimation Methods

Prediction of software development effort using Artificial Neural Networks has focused mostly on the accuracy comparison of algorithmic models rather than on the suitability of the approach for building software effort prediction systems. The use of back propagation learning algorithms on a multilayer perceptron in order to predict development effort is well described by Witting and Finnie [7]. Karunanithi [9] reports the use of neural networks for predicting software reliability; including experiments with both ABE and GA Neural networks. Samson [10] uses an Albus multiplayer perceptron in order to predict software effort. Nasser Tadayon [6] also reports the use of a neural network with a back propagation learning algorithm. However it is not clear how the dataset was divided for training and validation purposes. Khoshgoftaar [1] presented a case study considering real time software to predict the testability of each module from source code static measures. They consider Artificial Neural Networks as promising techniques to build predictive models. Finally in the last years, a great interest on the use of Artificial Neural Networks has grown. Artificial Neural Networks have been successfully applied to several problem domains. They can be used as predictive models because they are modeling techniques capable of modeling complex functions.

## IV. Methodology

Genetic Algorithm (GA) is a stochastic global optimization technique initially introduced by Holland in 1970's [6]. By mimicking biological selection and reproduction, GA can efficiently search through the solution space of complex problems and it is naturally parallel and provides opportunity to escape from local optimum. GA has become one of the most popular algorithms for optimization problems. In this section, we construct the OCFWANN system (stands for optimal projects of predicted Class and Feature Weighting and Artificial Neural Network

### A. Neural Network Based Cost Estimation.

The neural network architecture for software cost estimation is given as

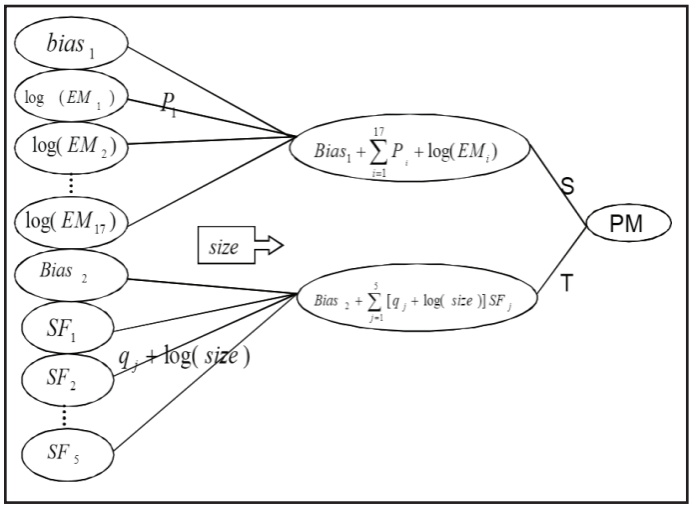


Fig. 1: Network Architecture

We compute the effort (PM) using the mathematical approach given by Tadayon[10].

Genetic Algorithm (GA) is a stochastic global optimization technique initially introduced by Holland in 1970's [6]. By mimicking biological selection and reproduction, GA can efficiently search through the solution space of complex problems and it is naturally parallel and provides opportunity to escape from local optimum. GA has become one of the most popular algorithms for optimization problems. In this section, we construct the OCFWANN system (stands for Optimal projects of predicted Class and Feature Weighting and Artificial Neural Network based Estimation) which can perform simultaneous optimization of 'N' projects of the predicted class and their feature weights. GA is selected as optimization tool for OCFWANN system.

### B. GA for Optimization

The procedure for OPFWANN system via Genetic Algorithm is presented in this section. The system consists of two stages: the first one is training stage and the second one is testing stage.

In the training stage 93 NASA data points are presented to the system, the ANN is configured with cost drivers to produce the cost prediction for the given input data point. The class label of the input data is determined basing on PM obtained. GA explores the class space to minimize the error (in terms of MMRE) of ANN on the training projects by the following steps:

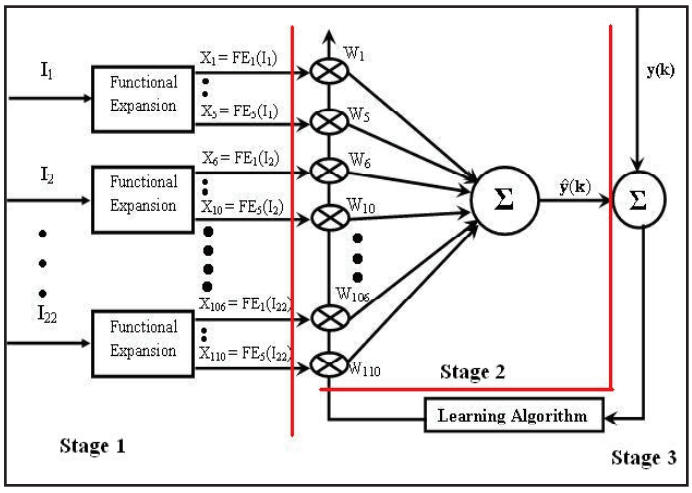


Fig. 2: GAFLANN Optimization

## V. Performance Evaluation Metrics

To measure the accuracies of the proposed methods, three performance metrics are considered: Mean Magnitude of Relative Error (MMRE), Median Magnitude of relative error (MdMRE),and PRED (0.25), because these measures are widely accepted in literature [6].

The MMRE is defined as:

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE \quad (2)$$

$$MRE = \left|\frac{(E_i - \hat{E}_i)}{E_i}\right| \quad (3)$$

Where n denotes the total number of projects, Eidenotes the actual cost of ith project, and denotes the estimated cost of the ith project. Small MMRE value indicates the low level of estimation error. However this metric is unbalanced and penalizes overestimation more than underestimation. The MdMRE is the median of all the MREs.

MdMRE = Median (MRE) (4)

It exhibits similar pattern to MMRE but it is more likely to select the true model especially in the underestimation case since it is less sensitive to extreme outlier [6]. The PRED (0.25) is the percentage of prediction that fall within 25 percent of actual cost

$$PRED(q) = \frac{k}{n} \quad (5)$$

Where n denotes the total number of projects and k denotes the number of projects whose MRE is less than or equal to q. Normally, q is set to be 0.25. The PRED(0.25) identifies cost estimations that are generally accurate, while MMRE is biased and not always reliable as a performance metric. However, MMRE has been de facto standard in the software cost estimation literature. Thus MMRE is selected for the fitness function in GA. More specifically, for each combination of 'N' project parameters and cost driver weights, MMRE is computed across the validation dataset. Then GA searches through the project parameter space to minimize MMRE.

### A. Encoding

To apply GA for searching, the cost drivers are encoded as binary string chromosome. Each individual chromosome consists of number of binary digits. There are six features for each driver (very low- vl, low-l, nominal-n, high-h, very high-vh, extra high-xh). Here we encode cost driver weights with 3 bits (0-n,1-vl,2-l,3-n,4-h,5vh,6-xh,7-n). 0 and 7 are assumed default values nominal (n). Since the cost driver weights are decimal values, before entering into ANN model these binary codes are transformed into decimal numbers.

### B. Population Generation and Fitness Function

After encoding the individual chromosome, the system then generates a population of chromosomes. Each chromosome is evaluated by the fitness function in GA. Since the GA is designed to maximize the fitness value and the smaller MMRE value indicates more accurate prediction, we set the fitness function as the reciprocal of

$$f = \frac{1}{MMRE} \quad (6)$$

Given one training project as input, ANN predicts the PM for the project, basing on the person month, the class is identified for the project, which contains set of similar projects as input. To evaluate the prediction performance of the ANN model, the error metric MMRE, PRED (0.25), and MdMRE applied on the training project set in the class. Then, the reciprocal of MMRE is used as the fitness value for each cost driver combination (or chromosome).

### C. Rules for Selection, Extinction and Multiplication

The standard roulette wheel is used to select chromosomes from the current population. The selected chromosome were consecutively paired with a probability of 0.8 was used to produce new chromosome in each pair. The newly created chromosome constituted a new population. The population is evolved by GA algorithm using evolutionary rules described above. The individual with best fitness value is in the population in every cycle.

### D. Completion of Evaluation

The population is evolved by the GA algorithm in the first stage until the number of generations is equal to or excess 2000 or the best fitness value did not change in the last 200 generations. The second stage is the testing stage. In this stage the system receives the optimized parameters from the training stage to configure the ANN model. The optimal ANN is then applied to the testing project to evaluate the performance of the trained ANN.

### E. Analogy-Based Estimation

Shepperd and Schofield introduced the ABE as a non-algorithmic method in 1997, and during the past few years, it has been widely used in the domain of estimation because of its simplicity and flexibility [3]. Its structure and overall steps is observed in Figure 3. In ABE method, the amount of effort for a software service is obtained from the former completed services, which we refer to this collection as the historical services collection. In fact, in this method, the features of these two services are compared one by one, and a distance is defined as the difference between these two services; this distance is obtained from Equation 3, and it is

called similarity function [5]. In order to measure the similarity, various formulas including Euclidean distance (Eq. 3), Manhatan distance (Eq. 4), and Grey have been suggested.

$$Sim(p,p') = \frac{1}{\left[\sqrt{\sum_{i=1}^{n} w_i Dis(f_i,f_i')} + \delta\right]} \delta = 0.0001$$

$$Dis(f_i,f_i') = \begin{cases} (f_i - f_i')^2 & \text{if } f_i \text{ and } f_i' \text{ are numerical or ordinal} \\ 0 & \text{if } f_i \text{ and } f_i' \text{ are nominal and } f_i = f_i' \\ 1 & \text{if } f_i \text{ and } f_i' \text{ are nominal and } f_i \neq f_i' \end{cases}$$

fi and f ′ i are independent features of the two services p and p′, and n is the total number of all the features of the services under study. W in the previous equation is the weight of every feature, which in fact shows the importance of that independent feature and its effect on the amount of the final effort. The Manhatan formula is very similar to that of Euclidean distance, but it computes the absolute difference between the features. Equation 4 shows the Manhatan similarity function [4].

$$Sim(p,p') = \frac{1}{\left[\sum_{i=1}^{n} w_i Dis(f_i,f_i') + \delta\right]} \delta = 0.0001$$

$$Dis(f_i,f_i') = \begin{cases} |f_i - f_i'| & \text{if } fi \text{ and } f_i' \text{ are numerical or ordinal} \\ 0 & \text{if } fi \text{ and } f_i' \text{ are nominal and } f_i = f_i' \\ 1 & \text{if } fi \text{ and } f_i' \text{ are nominal and } f_i = f_i' \end{cases}$$

In the normal sense and ABE0 version, one weight and Euclidean distance is used for all the features. The value of δ = 0.0001 is a small constant for inhibiting the denominator to be zero. The important point is that before comparing these two services, all the features must be normalized in order to obtain the steady changes in compared efforts. ABE measures the distance between two services by the independent and normalized variables between 0 and 1 of each service. The more the distance, the lesser the two services similarity. The solution function combines the obtained values from the earlier stage in a way that the amount of the final effort is obtained; in this function, the number of similar services for considering in the solution function is also determined (the value of the K Nearest Neighborhood variable (KNN)). The KNN value indicates the number of analogies and similar services that they show themselves in the solution function through the ABE method. Different researches have been conducted on the appropriate amount of K taken to be generally seen as desirable in the range of 1 to 5. However, the appropriate value depends on the type of data and dispersion. In fact, using different values and functions in ABE method creates different configurations that will naturally have different performances. In the experimental results section, different types of these values will be studied.

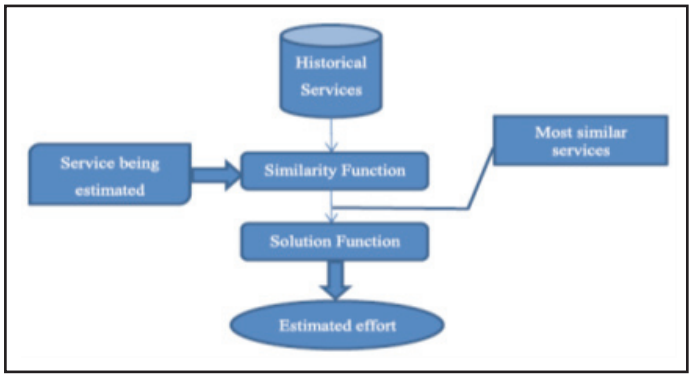


Fig. 3: Analogy Based Estimation Diagram

## VI. Results and Discussion

Table 1: Comparison of Error rate in Different Neural Networks

| RATE | ANN | FLANN | GAFLANN | ABE |
|---|---|---|---|---|
| MMRE(%) | 0.198405 | 0.182123 | 0.172378 | 1.160952e-2 |
| MDMRE(%) | 0.198930 | 0.182645 | 0.172092 | 9.098291e-2 |
| PRED(%) | 1.000000 | 1.00000 | 1.000000 | 9.024390e-2 |

## VII. Conclusion

Estimating software cost is a major for both the software industrial and academic researchers. In this project a comparative study is made based on four estimation techniques ANN, FLANN, GA-FLANN, ABE, in which ABE is proved to be the best technique in terms of reduced error rate within less time. The comparison results show that FLANN execution is very fast compared to ANN but error rate for the cost is more in FLANN. When compared FLANN to GA-FLANN, GA-FLANN with less error rate but estimation time is more. Hence both error rate and estimation time is proved to be best in ABE.

## References

[1] J. Herbsleb,"Global software engineering: the future of socio-technical coordination," Future of software engineering (FOSE'07), pp. 23-25, 2007.
[2] E. Conchuir, H. Holmstrom, P. Agerfalk, Brian. "Fitzgerland. Exploring the Assumed Benefits of Global Software Development," ICGSE'06, pp. 159-168, 2006.
[3] A. Gabriela,"Of Deadlocks and People ware - Collaborative Work Practices in Global Software Development", International Conference on Global Software Engineering (ICGSE'07), pp. 91-102, 2007.
[4] S. Bikram, C. Satish, S. Vibha,"A research agenda for distributed software development," International conference on software engineering (ICSE'06), pp. 731-740, 2006.
[5] D. Damian, D. Moitra,"Global Software Development: How Far Have We Come?", IEEE software, Vol. 23, No. 5, pp. 17-19, 2006.
[6] L. Ansgar, M.Jurgen,"Estimating the Effort Overhead in Global Software Development," International conference on Global Software Engineering, 2010.
[7] M. Fabriek, M. van de Brand, S. Brinkkemper, F. Harmsen, R. W. Helms,"Reasons for success and failure in offshore software development projects," European Conference on Information System, pp. 446-457, 2008.
[8] T. Carter, Cheaper's not always better. Dr. Dobb's Journal, March 1, [Online] Available: http://www.ddj.com/184415486 (accessed at Feb 25 2010),2006.
[9] G. Seshagiri, GSD: Not a business necessity, but a march of folly. IEEE Software, Vol. 23, No. 5, pp. 63-64, 2006.
[10] M. Ruchika, J. Ankita,"Software Effort Prediction using Statistical Machine Learning Methods," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 1, 2011.
[11] Boehm, B. W.,"Software Engineering Economics", Prentice-Hall: Englewood Cliffs, N. J., 1981.
[12] Putnam, L.H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Transactions on Software Engg., Vol. 4, No. 4, pp. 345–361, July 1978.
[13] Albrecht, A.J. J.R. Gaffney,"Software function, source lines of code, and development effort prediction a software science validation", IEEE Trans. on Softi. Eng., 9(6), pp. 639-648, 1983.

J. Sruthi pursuing her M.Tech in the department of Computer Science and Engineering, Raghu engineering college, Dakamarri Village, Bhimunipatnam Mandal, Visakhapatnam, A.P., India. Affiliated to Jawaharlal Nehru Technological University, Kakinada. Approved by AICTE, NEW DELHI. She obtained her B.Tech (CSE) from Dadi Institute of Engineering and Technology, Visakhapatnam.

Dr. Jose Moses working as Associate Professor and Head of the Department in the department of Computer Science and Engineering, Raghu Engineering College, Dakamarri Village, Bhimunipatnam Mandal, Visakhapatnam, A.P., India. Affiliated to Jawaharlal Nehru Technological University, Kakinada. Approved by AICTE, New Delhi. He published several papers inNational and International Journals. He completed his M.Tech(CSE) and Ph.D (CSE). He is active member of variousprofessional bodies. His research interests lies in Computer Networks.

M Krishna Kishore working as Associate Professor in the department of Computer Science and Engineering, Raghu Engg. college, Dakamarri Village, Bhimunipatnam Mandal, Visakhapatnam, A.P., India. Affiliated to Jawaharlal Nehru Technological University, Kakinada. Approved by AICTE, NEW DELHI. He published several papers in National and International Journals. He obtained his M.Tech(IT) from Andhra University and pursuing Ph.D (ECE)in Gitam University. He is active member of various professional bodies. His research interests lies in Under Water Communication.