



# Preliminary performance study of a brief review on machine learning techniques for analogy based software effort estimation

K. Harish Kumar<sup>1</sup> · K. Srinivas<sup>1</sup>

Received: 28 November 2020 / Accepted: 5 August 2021 / Published online: 23 September 2021  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

## Abstract

Analogy-based effort estimation is the major task of software engineering which estimates the effort required for new software projects using existing histories for corresponding development and management. In general, the high accuracy of software effort estimation techniques can be a non-solvable problem we named as multi-objective problem. Recently, most of the authors have been used machine learning techniques for the same process however not possible to meet the higher performance. Moreover, existing software effort estimation techniques are mostly affected by bias and subjectivity problems. Analogy based effort estimation (ABE) is the most extensively conventional technique because of its effortlessness and evaluation ability. We define five research questions are defined to get clear thoughts on ABE studies. Improvement of ABE can be done through supervised learning techniques and unsupervised learning techniques. Furthermore, the results can be knowingly affected by the different performance metrics in ABE configuration.

**Keywords** Machine learning · ABE · Software effort estimation · Supervised techniques · And unsupervised techniques

## 1 Introduction

In the case of global software development (GSD), analogy software effort opinion is the effort necessary to develop a global software plan (Peixoto et al. 2010). Development efforts are seen as an important factor in programmatic factors, especially in the case of global growth, which is very difficult to predict (Lamersdorf et al. 2014). More than the history few years, GSD has provided a figure of ways to assist project managers in developing project software. In particular, database-based methods are used by historical data applications to assess the effort required in a newly distributed software program (Faria and Miranda 2012). Linear regression and case-based logic are the most common methods of calculation. This data includes information on cost carriers, related components, and projects actually implemented. The data generation method of the equations based on estimates is usually the most expensive driver interpretation used to estimate the effort required for a new project

(Jørgensen 2004; Grimstad and Jørgensen 2009; Singh and Misra 2012; Boehm et al. 1995). The main objective of this work is to expand software-relevant techniques for GSD schemes and to enable managers to implement and estimate specific changes quickly and accurately. In short, the purpose of this work is Khatibi and Jawawi (2011), Boehm et al. (2000).

- Learn how project software can integrate project management with the GSD method.
- Progress a cooperative mechanism to imagine the influence of cost carriers on GSD schemes.
- Create device support for the hardware (El Bajta 2015).

Evaluate software development efforts is an important task in software project organization. As a result, he has attracted the attention of researchers and has recommended a number of software development technologies over the past three decades (Mendes 2008). This technology is divided into 2 categories:

1. Models of parameters derived from arithmetical or numerical analysis of historical project data;
2. Non-parametric models based on synthetic neural networks, genetic algorithm, regression tree, rule-based

✉ K. Harish Kumar  
khrsharma@gmail.com

<sup>1</sup> Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Deemed to be University, Hyderabad, Telangana 500075, India

induction, analogy or CBR, artificial intelligence and fuzzy logic (Kumar et al. 2008a, b).

Analogy based analysis attempts good technology appeared in the evaluation field in the above models. Analog assessment is a case-based logic designed to solve a new problem using the results of a preliminary procedure. Similarly, evaluation of effort has two chief compensation (Huang et al. 2008). First, it is an instinctive way for users to understand and explain (as opposed to black box approach such as neural networks); Second, complex relationships with dependent variables (such as effort or cost) and sovereign variables (cost carriers) can be used as a model (Elish 2009). However, analogy-based effort estimation (ABE) methods have some limitations. First, they are partial by the incapability to address lost values (Stefanowski 2006). Second, they are perceptive to inappropriate and undesirable features when evaluating the extent of crash of an attribute. Third, they cannot knob properties previous than binary variables. As a result, they do not work properly with the database of software projects with certain features. This article focuses on the purpose of different types of properties (Shepperd and Schofield 1997).

Classification attribute are typically evaluate with linguistic morals such as "inferior", "complex", and "significant". These linguistic meanings are derived (or not) from arithmetical meanings (Ahmed and Muzaffar 2009). If they are copied from arithmetical data, they are often called traditional interval (e.g., years of usage knowledge are deliberate and software dependability is calculated based on the digit of defects) (Azzeh et al. 2009). Such an understanding does not reflect the interpretation of human language values, and as a result can lead to misunderstandings and uncertainties (Mendes et al. 2002). To solve this problem, you suggested a new method in the previous work to evaluate the similar effort of the fuzzy analogy, which combines the fuzzy logic with the analogy logic. In obscure analogues, classical spaces represent linguistic meanings rather than obscure sets (Amazal et al. 2014a, b). The mechanism explores an attempted evaluation method based on similarity. The basic

idea of the analog effort evaluation system is that similar programs typically use approximate development efforts. Based on this assumption, one or more historical projects can evaluate a new project initiative. Comparable past project are famous by measure the similarity between the projects and the past project, and the magnitude of the similarities frequently determines the Euclidean coldness between the scheme. The working process of ABE consists of 5 basic steps:

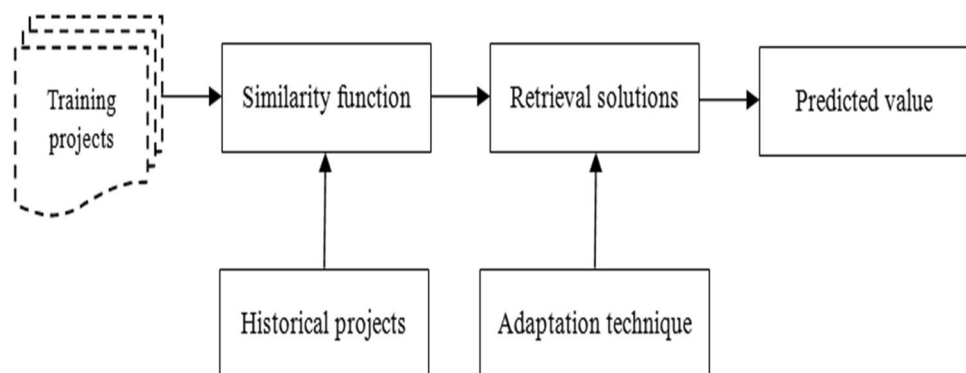
1. Choose the database of historical projects.
2. Select Project description (i.e. cost drivers) for the integration function.
3. Quantify the resemblance among the new plan (i.e. the target plan) and the historical plans.
4. Recognize historical projects (or analogues) comparable to the target project.
5. Study a comparable effort to make an effort estimation for the Target Project (Wen et al. 2009).

## 2 Related works

Over the past decade, the difficulty of estimating software development project has attracted the attention of researchers, so innovative methods of machine learning are being used in this field (Nitze et al. 2014; Walkerden and Jeffery 1999). Most estimation in this area uses the research observation method because they contain a variety of factual data, including information about previously completed project plans. Optimization methods, neural networks, and blurry method are extensively used in the estimation of software expansion efforts (Chiu and Huang 2007; Azzeh et al. 2011). Figure 1 shows the general framework of analogy-based estimation.

Idri et al. (2016) Data Technology does not use two analogy software development ratings: Classic Analogy and Fuzzy Analogy. Added particularly, we analyze the prognostic routine of these two analogy technology with tolerance, elimination, or KK- nearest Neighbourhood (KNN) stimuli.

**Fig. 1** General structure of analogy-based software effort estimation



Seven data sets, three MD. MD percentage range from 10 to 90%. M.D. marks show that fuse analogise produce more true estimate based on standard accuracy measurement (SA) than classical analogues, regardless of hardware, data set, method, and MD proportion. Furthermore, this study showed that the use of KNN stimuli instead of tolerance or exclusion may get better the predictive correctness of analogy methods. In addition, tolerance, exclusion, and KNN stimulation affect the percentage of lost algorithm and MD, which significantly affects forecasting efforts.

Moosavi and Bardsiri (2017) present a model based on the adaptive neuro-fuzzy interference system (ANFIS) and satin bowerbird optimization system (SBO) for more correct software evaluation. SBO is a novel optimization algorithm planned to solve ANFIS mechanism using minor and reasonable modifications. The planned hybrid mould is the optimal neuro-fuzzy evaluation mould that can generate accurate assessments across a broad variety of software. The planned optimization algorithm can be compare with other biodegradable optimization algorithms that include similar and multimodal functions using 13 standard test functions. In addition, the planned hybrid model is evaluated with three accurate data.

Dragicevic et al. (2017) presented to predict any active mode. With simple, small, collection methods, the recommended sample speed will have no practical effect. During the planning phase, this model can be used quickly. The authors define the structure of a particular model and automatically evaluate the parameter in the database. Get data from the software company's saturated active programs. This paper describes the various statistical data used to calculate sample accuracy: relative error, level d forecast, accuracy, absolute error, average square error, relative absolute error, and relative double error.

Qi et al. (2017) aim to gather enough data to create a model for assessment efforts to address learning data deficits. It offers my GitHub to collect enough and different original data to evaluate the effort. A measurement system is based on GitHub data to evaluate employee measurements. An update model approach is based on the adaboost and categorization and regression tree (ABACRT) to dynamically develop the composed database.

Jørgensen (2016) presented software that usually based on expert evaluation, and the actual use of the effort is less than reflective. The main task is to understand how the selection of the force unit affects the evaluation of expert-based efforts and to use this knowledge to improve the reality of the evaluation of efforts. This method involves two tests, which require software professionals to simultaneously evaluate project effort on working methods or working days.

Zare et al. (2016) present software initiative estimation model based on the size of the three components of the Bayesian Network and the 15 mechanism of COCOMO and the

software. Toxin works at specific intervals for network nodes. Optimal update coefficient of opinion based on the notion of finest manage for optimizing the COCOMONASA database genetic algorithm and particle mass. In previous words, the sketchy value of the effort varies depending on the optimal coefficient. To evaluate the software effort, taking into account the value of the identified software and eliminating the number of defects, according to the three stages of the requirements, if the specificity of the features, plan and code exceeds the specified number of defects. This article is a review of fresh study to get better the effectiveness of comparative evaluation. The marks of this study were very useful for the researchers to understand the features evaluated in the previous studies and to increase the accuracy and consistency of the comparisons obtained. The main issues identified for further investigation as a result of the current investigation are technology, quantity, structure, and estimation.

### 3 Analogy based estimation (ABE)

Shepperd and Schofield suggested the ABE model instead of the algorithm model. In this model, development efforts are evaluated through the comparison process of selecting a new project and a similar project. The selected project effort is used to evaluate the new project effort. ABE is extensively worn in software growth and evaluation due to its simplicity and evaluation capability. ABE basically consists of four components (Bardsiri et al. 2014).

- (i) Historical dataset to develop dataset previous project information has to be collected.
- (ii) Similarity function—For comparison purpose attribute such as FP and LOC are chosen.
- (iii) Retrieval rules—Similar to new projects previous projects is retrieved.
- (iv) Solution function—The effort of new scheme is approximate.

#### 3.1 Similarity function

ABE comparison occupation is used to determine the comparison by comparing the characteristics of the two circuits. There are two accepted function of unity: the Euclidean similarity ( $S_1$ ) and the Manhattan similarity ( $S_2$ ). Euclidean similarity function (Angelis and Stamelos 2000) is given as follows:

$$S_1(p, p^{-1}) = \frac{1}{\sum_{i=1}^n w_i D(f_i, f_i^{-1}) + \partial} \quad (1)$$

$$D(f_i, f_i^{-1}) = \begin{cases} (f_i, -f_i^{-1}) \text{ both are nominal} \\ 0 \text{ nominal } f_i = f_i^{-1} \\ 1 \text{ nominal } f_i \neq f_i^{-1} \end{cases} \tag{2}$$

where  $p$  and  $p^{-1}$  are comparison project and  $w_i$  are the heaviness assign to every property. Weight vary from 0 to 1. Additionally,  $f_1$  and  $f_0$  show inequality of each circuit and the number of  $n$  attribute. Nothing is used to get zero consequence. The MS method is alike to ES, but compute the completed is similarity among the attribute. The Manhattan uniform purpose is represents as follows:

$$S_2(p, p^{-1}) = \frac{1}{\sum_{i=1}^n w_i D(f_i, f_i^{-1}) + \partial}, \quad \partial = 0.0001 \tag{3}$$

$$D(f_i, f_i^{-1}) = \begin{cases} (f_i, -f_i^{-1}) \text{ both are nominal} \\ 0 \text{ nominal } f_i = f_i^{-1} \\ 1 \text{ nominal } f_i \neq f_i^{-1} \end{cases} \tag{4}$$

The other similarities functions used for different machine learning algorithms are average equivalence rank, maximum distance equivalence, and Minkowski equivalence used in previous studies (Nisar et al. 2008).

### 3.2 Solution function

Explanation features are used to evaluate software growth efforts by identifying alike project as similar processes. Features of popular solutions: similarity of similar projects, standard of similar project, average of alike project, average distance weight average. Here,  $n$  represents the average effort for similar projects (Kadoda 2000).

$$S_p = \sum_{i=1}^n \frac{S_1(p, p_i)}{\sum_{i=1}^n S_2(p, p_i)} S_{pi} \tag{5}$$

If  $p$  represents a new system,  $p_i$  represents a program similar to the return,  $S_{pi}$  is a test value of the same program as the return,  $S_1(p, p_i)$  is the resemblance among the  $p$  and  $p_i$  systems, and  $K$  is a similar project overall in previous studies. Various functions of the solution have been used. Many studies use single one answer purpose, while other study uses different type of remedial action (Li and Ruhe 2008).

### 3.3 Machine learning techniques for ABE

In current duration, software development effort estimation (SDEE) research has inward more attention from machine learning (ML) based methods. Some researchers consider the ML-based means to be one of the three main methods of evaluation. Boehm and Sullivan considered learning methods to be one of six types of programmatic effort evaluations (Wen et al. 2012). Zhang and Tsai abridged the use of a variety of ML technologies in the SDE industry, counting case-based evidence, findings, artificial neurological networks, and genetic algorithms.

This review confirms the lack of experiential research on the use of BN, SVR, GA, GP, and AR technologies (Zhang and Tsai 2003). Consequently, researchers are requested to demeanour more experiential research on the hardly ever used ML technology, which additional strengthens the experiential proof for its effectiveness. In addition, ML urges researchers to explore the possibility of using outdated hardware to evaluate software development efforts. Researchers have closely monitored related areas such as mechanism knowledge, data processing, figures and false cleverness to more effectively detect and use outdated ML technologies (Mair et al. 2000). High quality recorder data designed with thorough images of project description and data album processes are important for building and validate ML models (Alpaydin 2004). Aside from this review, most existing databases are out of date and the number of projects in this database is very little. On the other hand, the project database is hard to gather and uphold and usually contains secret in order. To solve these trouble and facilitate SDEE investigate, researchers are advised to share ownership scheme data with the research neighbourhood once privacy has been achieved. The PROMISE collection is a opportune place to share SDEE data sets when needed (Bibi and Stamelos 2006).

## 4 Review methodologies

We planned and conducted the review in accordance with the procedures recommended by Kitchenham and Charters. There are six main steps in this configuration, as shown in Fig. 2. First, a number of research questions are set out based on what needs to be done for current research. Second, explains the search strategy to show how related studies are defined. Search terms and sources are defined in this space. Third, it includes modified research selection criteria that cannot be used to answer investigate question. Fourth, selected studies based on quality characteristics will be further refined. Finally, the necessary in sequence is obtained from certain study and analyzed

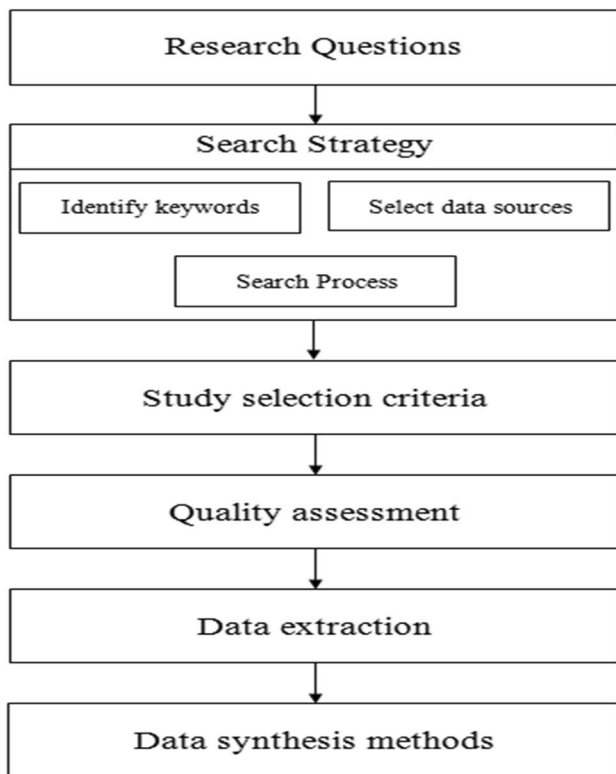


Fig. 2 Overview of review methodology of article

according to the investigate question. particulars of every step are given in the subsequent section.

#### 4.1 Research questions

The aim of this examination is to analyze study that focus on developing an analogy based effort estimation model for evaluating software project. Discussion of review questions leads to a capable review methodology process. The main object of the review is to answer the following investigate question as follows:

- RQ1. How did ABE improve in previous study?  
 RQ2. How is the correctness of the model recommended in previous study calculated?  
 RQ3. How the machine learning (ML) technique were is compared to the other previous models?  
 RQ4. Did we get the same attention from researchers at different stages of the analogy process?  
 RQ5. Which mechanisms of ABE (resemblance and explanation function) were used in previous study?

#### 4.2 Search strategy

The goal of the search scheme is to find research that will help RQ respond. The three steps of a search strategy are to

identify keywords, define search strings, select data sources, and finally search for data sources.

##### 4.2.1 Identifying keywords and defining search strings

We used the following terms to derive search terms (Ezghari and Zahi 2018):

- Identify the keywords that match the question scheduled over.
- Find all synonyms and spell of vocabulary.
- Use a Boolean operative to get any record of the terms (or all) OR to join in identical conditions.
- In order to connect the main terms the Boolean operator AND should be used and to record any evidence including all the conditions

The entire set of investigate terms was prepared as follow: (analogy OR “analogy-based reasoning” OR “case-based reasoning” OR CBR) AND (software OR system OR application OR product OR project OR development OR Web) AND (effort OR cost OR resource) AND (estimat\* OR predict\* OR assess\*).

##### 4.2.2 Selected data sources

The database contains 7 electronic databases (SpringerLink, IEEE Xplore, ACM Digital records, Elsevier Science Direct, CiteSeer, InderScience, and Google Scholar) for preliminary research. Some other important sources, such as DPLP, SiteSear, and Computer Science Bibliography, will not be considered, as they are fully covered by selected data sources. The search terms in 8 web databases before searching for journal and conference papers. This is because different database search engines use different search strings to sort different databases. Search the top five databases of titles, abstract, and keywords Google Scholar offers full text search with a single title and millions of irrelevant entries. We have limited search from 2000 to 2020.

##### 4.2.3 Search process

*Search phase 1* Search eight databases independently, then collect return documents from BEST web and create a set of applicant pass.

*Search phase 2* scrutinize the relevant paper orientation lists to find the most pertinent paper If so, add them to the package.

#### 4.3 Selection criteria

The systematic search time of the review is not ideal as the review may be biased in selecting relevant studies. At this

stage, the content of the paper, the abstract, and the conclusion of the conference should be evaluated to select the objectives of the current research. The addition and subtraction criteria are defined as follow.

Enclosure:

- ABE considered being the major replica of evaluation.
- In the case of duplicate documents, the newest and most complete sheet is selected.
- ABE improves the performance (not only using ABE as a balancing model).

Exclusion:

- Evaluate software measurements in addition to efforts such as time and measure.
- In addition to development efforts such as testing and technical effort, evaluate the value of the amount of effort.
- Study topic is software scheme control.
- Duplicate publication of the same study (there are several publication of the same study, only the most comprehensive in the review).

This study evaluates software development efforts and focuses only on studies that have worked to improve ABE performance. Therefore, the aims and objectives of every paper are cautiously considered according to the assortment criteria, which will lead to a review of the certain studies.

#### 4.4 Quality assessment

A selected study in quality assessment (QA) is first used to identify data obtained in meta-analysis (Huang et al. 2017), this is an significant data collection scheme. though, the data found in this reconsider are made in dissimilar test formats, and meta-analysis is not used if the quantity of data is comparatively little. For this reason, quality evaluation results are not used to obtain quality. Instead, we used it to interpret review results and show the power of assumptions. In addition, quality evaluation results are provided as additional selection criteria. Possible answers to this question are as follows: We have divided the excellence appraisal question to evaluate the intensity, reliability and significance of the pertinent study. These question are listed in board 1 and are derived from some of them (e.g. QA2, QA3, QA5, and QA6) (Idri et al. 2016). There are only three answers to each question: "Yes", "Partly" or "No". All three answer were rated as follows: "Yes" = 1, "Partly" = 0.5, "No" = 0. For this cram, its rating was calculated instead of QA.

The aim of this study is to assess examine study that get better the ability to assess the ABE means [QA1]. a quantity of previous study have used ABE as an additional

technology to provide a overweight model. These types of study do not help to react the investigate question of the current study. Consequently, in the first phase of excellence assessment, we identified the main objectives for improving ABE.

#### 4.5 Data extraction

At this stage, data were collected for research questions and activities before selecting relevant questions. Each study is carefully analyzed to reach conclusions and to obtain useful data to response previous research question.

#### 4.6 Data synthesis

The principle of the database is to gather confirmation beginning positive study to answer examine question. Once the data collection is complete, the analysis and review process will be simplified and the results will be expanded. At this stage, the selected files are compiled and integrated, which simplifies the comparison process between different types of papers. The data collected are briefly listed base on the key factors used to speak to investigate questions. Data collection is an important part of methodical reconsiders because evidence and conclusion can be drawn as a result of precise examination of data composed from previous study.

#### 4.7 Selected papers

Figure 3 represents the general method of selecting paper. 410 papers were found in all search engines. Deleting duplicate files will reduce the number of files to 201, and selecting the criteria will reduce that number to 76 pages. Finally, the quality assessment of the remaining papers provided a total of 32 papers.

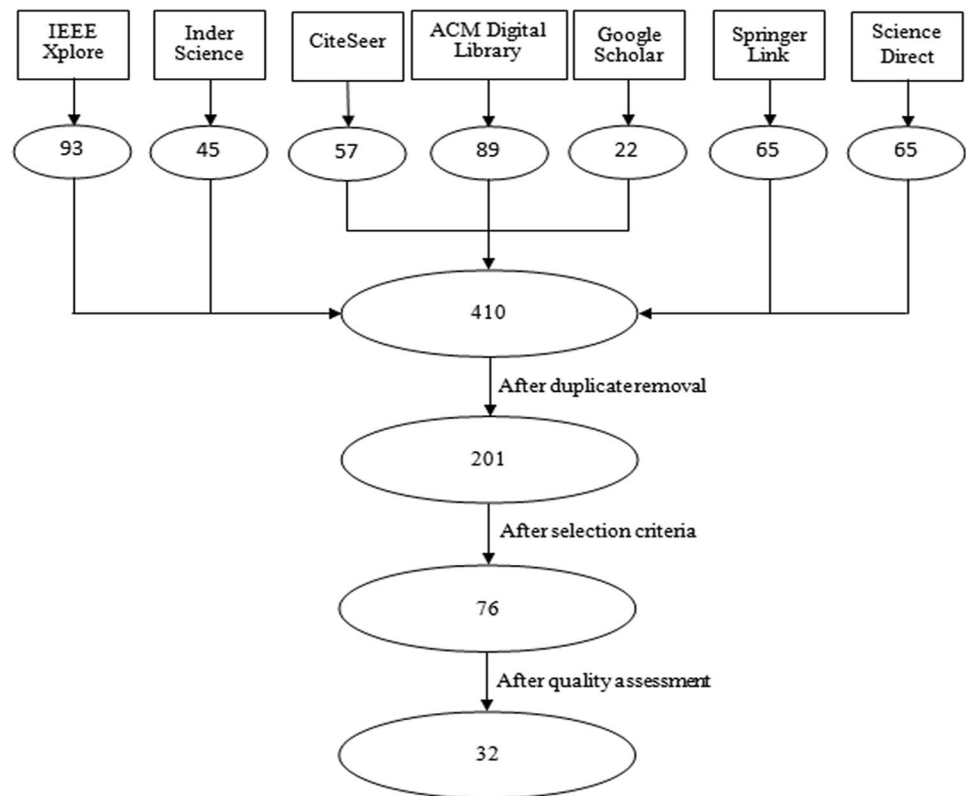
### 5 Machine learning algorithm for analogy based software effort estimation

Machine learning is often used successfully to solve problems, improve system performance and improve machine design. Any example in a database is the use of machine learning methods with similar characteristics. Contrary to the installed study, if known labels represent events, it is called supervised learning, in dissimilarity to unsupervised learning, where illustrations are unlabelled.

#### 5.1 Supervised machine learning

The classification of supervision is one of the most commonly performed tasks by intelligent systems. Therefore, a number of technique have been urbanized base on artificial

**Fig. 3** Selection and filtering procedure of research papers from different search engines



intelligence. In this section, we will focus on the most important techniques in machine learning, starting with classification-based technologies.

### 5.1.1 Classification based techniques

Several ML programs contain tasks that can be setup as supervised. In particular, this classification of works is related to problems that identify only the values of the results of events that are different and sorted. Classification based techniques consists of several techniques such as support vector machines, naive Bayes, nearest neighbours.

**5.1.1.1 Support vector machines** Gupta et al. (Sikka and Verma 2011) have studied that most effort evaluations are based on method or analogy. COCOMO 81 and Functional Points are the most popular algorithm models. Similarly, estimation is basically a case-based logic. By analogy, machine learning techniques such as obscure logic, gray matter theory, genetic algorithm, and subvector machines are used to improve predictions. Based on these similarities, we will explore software effort evaluation approaches and accurately compare some of the technologies commonly used in terms of size. Shashank et al. (Satapathy and Rath 2014) have studied that support Vector regression (SVR) is one of the various modes of soft computing technology that allows you to achieve optimal values. The SVR concept

is based on the calculation of a linear regression function with high characteristic intervals representing a linear function of the input data. Additionally, SVR kernel methods can be used to modify the input data and to obtain the optimal boundary between the possible results based on these changes. The main goal of the scientific work conducted in this study was to evaluate the strength of the project using a class point method. Tirimula et al. (Benala and Bandarupalli 2016) various methods are suggested to correct the solution obtained and to improve the accuracy of ABE. According to the study, all available measurement method depends on linear regulatory methods excluding for linear and nonlinear schemes, which are calculated on the basis of a synthetic neural network. After proper testing of a superior calibration means, the slightest square support vector machine (LS-SVM) displays a confident beam that acts as a linear error correction means for computational calculations. In current studies, LS-SVM is used to improve ABE. Specific tasks will be explored in a database consisting of three parallel data sets compared to other artificial neural network (ANNs) and extreme learning machines (ELMs).

Lin et al. (2011) have studied that Attempting to miscalculate the program at an early stage can lead to disastrous results. It not only works on the spreadsheet but also adds value. This will lead to a huge deficit. Factors influencing project development also vary, as different program development groups have their own way of evaluating their own

efforts. To solve these problems, we suggest a prototype algorithm that combine genetic algorithm (GA) and support vector machines (SVM). The SVM can determine the greatest limitation at intervals and make accurate predictions according to a particular model. Pospieszny et al. (2018) have studied that gap between the results and implementation of an organization's innovative research by proposing approaches to the Introduce and uphold effective and practical machine knowledge using investigate innovations and industry best practice. This was achieved using three methods of ISBSG database, intelligent data processing, machine learning (vector machines, neural networks, common linear models), and cross validation. Oliveira et al. (2010) have proposed Mechanical hardware software initiatives such as radial base function (RPF) neural networks, multi-layer perceptron (MLP) neural networks, and vector regression support (SVR) based are Recently used for evaluation. Some documentation shows that the degree of accuracy of software effort evaluation depends on the parameter values of these methods. Furthermore, it shows that the choice of input features can significantly affect the accuracy of the assessment. (1) Select a subset of the genetic algorithm for optimal input behaviour, (2) improve the parameters of machine learning methods, and achieve greater accuracy in evaluating software effort. Corazza et al. (2011) have proposed that support Vector regression (SVR) is a new creation machine learning algorithm suitable for data forecasting modelling problems. The purpose of this paper is twofold: first, read the presentation of SVRs for evaluating web-based tasks using an enterprise database; second, compare dissimilar SVR configurations for better performance. Specifically, we combined three variables (pre-processing, normalization, and logarithm) with two different dependent variables (effort and reverse). As a result, SVR was used with six different data configurations. In addition, non-core SVRs were used to solve linear problems to understand the importance of key functions, and 18 different SVR configurations were obtained using radial base function (RBF) and different types of cores. We defined the identification procedure based on an integrated cross-test approach with values suitable for each configuration and each parameter.

**5.1.1.2 Naive Bayes** Shivhare et al. (Braga et al. 2007) presents the evaluation approach using mechanism knowledge technique for quantitative data and is implemented in two stages. The first step focuses on selecting the most approving feature for large amounts of data related to previous projects. Quantitative analysis is performed using the property reduction probabilistic theory. The second stage evaluates the exertion base on the optimal properties obtain from the first stage. Evaluation is done differently using naïve Bayes classifier and artificial neural network techniques. The first step is to review the pub-

lic domain data (USP05) for the asset reduction process. Specific methods evaluate and compare the mean magnitude of relative error (MMRE), the root mean square error (RMSE), the mean absolute error (MAE), and the correlation coefficient in terms of option. BaniMustafa (2018) suggests performing. The forecast includes 93 projects using three machine learning techniques used to pre-process COCOMO data for NASA's core data: naïve Bayes, logistics regression, and random forest. The generated samples were experienced using five cross-tests and evaluated using categorization accuracy, accuracy, performance, and AUC. The evaluation grades were compare with the COCOMO evaluation. All the techniques used succeeded in achieving better results than the comparable Kokomo model. However, now he was separated by speed and random jungle. While naïve Bayes performed better than the ROC curve and recruitment score, Random Forests had a more confusing matrix and performed better in classification and precision operations.

Zhang et al. (2017) have studied that embryonic option approaches to testing for growth toxicity is an important and urgent task in the drug development process. Innocent Pro classification is used in this study to create a new predictive model for this toxin. The established diagnostic model was evaluated with 5 internal cross-tests and one external examination. The internal diagnostic results of the study were 96.6% and 82.8%, respectively. In addition, it defines four simple explanations and augments the molecule with some representation of toxins. Therefore, we hope to use the established silicon prognosis model as an alternative means of toxicology assessment. This molecular information will help in in-depth understanding of toxins and their clinical chemists in drug diagnosis and lead optimization.

Wen et al. (Zhang et al. 2015) offer a bream algorithm to predict software effort and two embedded strategies for managing lost data. The MDT strategy ignores lost data when using BREM software for forecasting, and the lost data when the MDI strategy describes the forecast model using recorded data is considered functional.

Hussain et al. (2013) in addition, many companies have taken active steps in delivering software to teams designed to evaluate software components. Each repetition. Cosmic is an ISO/IEC international criterion that is an objective way of measure the size of software based on the needs of the user. At the grain level, where external connections with the computer are known to human dimensions, it needs to be edited and distorted by the cosmic user. This task of taking time with fast processes is avoided because it is the only way for quick subjective judgment in human matters. In this article, we will explore these issues, from the need for informally written text to the approach to the cosmic approximate functional scale, and demonstrate its application in popular dynamic processes.



**5.1.1.3 Nearest neighbours** Sarro and Petrozziello (2018) proposed a new way based on linear indoctrination (dubbed as linear programming for effort estimation, LP4EE) and conducted a comprehensive experiential study to evaluate the performance of LP4EE and ATLM.

Li et al. (2008) Most of the existing estimation methods generate only statistical data. Due to the inherent uncertainty and complexity of the technical process, it is often very difficult to get an accurate point estimate. Hence, some previous study have focused on probability prediction. Analogy based estimation (ABE) is a accepted technique for evaluating scores. This means is widely conventional because of its ideological plainness and competitiveness of experience. However, there is still a lack of probability structure for the APE model. The PABE forecast is obtained by synchronizing the Boeing result of the neighbouring K parameter. PABE tested four repairs more than any other installed evaluation technology.

Le-Do et al. (2010) the APE examined whether it promotes new project initiatives from historic project initiatives with like character. ABE is easy, although it can have an impact on historic projects. Noise is data corruption that adversely affects the presentation of a model built on historical data. In this study, we recommend a sound filtering approach to improve ABE accuracy in past projects. We recommend and Effort-Inconsistency Degree (EID) from similar programs as long as they are not used by historical program initiatives. We compile and filter sound based on EID and random historical project data. We tested ABE's functionality using three representative technologies of our approach and refinement, namely a modified neighbourhood algorithm, integrated weight reduction based on project selection, and a genetic algorithm based on three software project databases.

## 5.1.2 Regression based techniques

There are several applications for machine learning, the most significant of which is regression-based technique. Regression based techniques is further classified as decision tree and neural networks.

**5.1.2.1 Decision tree** Nassif et al. (2013) there is a perception that revaluation leads to discounts and financial losses in the business. There are several models of software testing; however, he did not become the best in all circumstances. In this study, decision tree forest (DTF) model was compared with the traditional decision tree (DT) and multiple linear regression model (MLR) models. The assessment was conduct using the ISBSG and Desharna is business databases. The marks show that the DTF model is reasonable and can be used as a substitute for software effort forecast.

Mohanty et al. (2010) many methods can be used, such as non-standard method, vague logic, final wood, and solid packaging used by NNSE. The review will be useful to researchers as a preparatory point as it offers significant research directions for the future. Coach reviews can be helpful. This will gradually lead to SE completion and provide better, more dependable and cheaper software goods.

Papatheocharous and Andreou (2009) the software tries to predict the problem with ambiguous results generated using historical project data models. In addition, they studied the taxonomic laws developed by various digital and nominal project properties that were predicted to be used in the software development process. The approach seeks to classify past project data into homogeneous clusters to ensure an accurate and reliable estimate of each cluster. CHAID and CART algorithms are used in data records of approximately 1000 project values, which are analyzed and pre-processed to generate ambiguous wood cases and estimate the accuracy of predictions obtained through production classification rules.

Trendowicz and Jeffery (2014) memory-based assessment method such as case-based logic does not create assumptions about the formation of the assignment crack relationship. Model-based approach, such as arithmetical weakening or recent perspectives, predetermines the specific structure of the effort model; however, in the case of a specific attempt, they do not power any precise event into the model.

**5.1.2.2 Neural networks** Idri et al. (Zakrani and Zahi 2010) many researchers are trying to create new models and improve existing models using reproduction intelligence techniques: case-based rationalization, decision-making perspectives, genetic systems, and neural networks. This paper is dedicated to the design of radial base operating networks for estimating software costs. This shows the influence of the RBFN network structure, mainly on the number of hidden layer neurons and the accuracy of production estimates in the MMRE and Pred indicators on the width of the stem function.

Kumar et al. (2008a, b) the use of wavelet neural network (WNN) is recommended for software development efforts. multilayer perception (MLP), radial basis function network (RBFN), Multiple Linear Rest (MLR), Dynamically Developing Neuroplaying System (Delphic), Auxiliary Vector Machine DB Average measurement relative error (MMRE) based on Data Processing Services (IBMTPS) database.

de Campos Souza et al. (2019) offer the use of an ambiguous neural network containing ambiguous rules, which will help to create a special system based on the described rules, which will help to predict the software progress time according to the difficulty of the program components. To prop up the data validation method, the growing solidity practice is recommended for the primary deposit of the sample. To

recover the compassion of neurons in the neural network, a leak-type common cement purpose is used to get marks.

Kaushik and Singal (2019) proposes a Algorithm for SDEE Non-hardware, e.g. Hybrid Model of wavelet neural network (WNN), Meteoritic Algorithm. It uses two modification systems, namely the Firefly algorithm and the bad algorithm. The work of the WNN is explored by integrating each of these transformation systems. Morley and Gaussian use two types of bandwidth functions as WNN activation functions. Specific devices are evaluated during the trial period in PROMISE SDEE collections. The integration of meteoritic algorithms outperforms the predictive results compared to the traditional WNN.

Huang and Chiu (2009) proposes a fuzzy neural network (FNN) An move towards that incorporates an reproduction neural network into vague conceptual process to obtain a software effort evaluation. An synthetic neural network is used to establish vague rules that are important in vague inference processes.

Rao and Kumar (2015) proposes a Use the Advanced Software Effort Evaluation for the Common Regressive Neural Network COCOMO Database. The mean magnitude relative error (MMRE) and Median Magnitude Relative Error (MdmRE) are used as assessment criterion in this paper. The specific global waning neural network compares the M5 with a mixture of technologies such as linear waning, SMO polysernel, and RBF kernels.

## 5.2 Unsupervised learning

Probabilistic Latent Semantic Analysis (PLSA) for power learning techniques aims to identify and discriminate exchanged contexts of word use without the aid of a vocabulary or vocabulary. It has at least two main effects: First, it allows the reduction of polysemis, i.e. words with multiple meanings, basically all words are a polysemy. Second, it expresses surface similarities by combining words that are part of a general context. As a special case, it includes similar words, i.e. words that are similar or have almost the same meaning.

### 5.2.1 Clustering technique

Although it is possible to create stream groups by writing code that knows the attributes, we found in the IAT / pocket size layers that this approach enables a high human interpretation of the results. Nor can it be sufficient to apply the method to different network types. Machine knowledge technique can be used to identify cluster data streams and generate cluster classifications. The various clustering techniques are Fuzzy, KNN, ANN, CNN and DNN.

**5.2.1.1 Fuzzy** Azzeh et al. (2008) the subcommittee explores the impact of sample algorithms on getting better the accuracy of the comparison software effort evaluation model. They recommend a subcommittee selection method based on vague logic for analog software attempt evaluation models.

Amazal et al. (2014a, b) in a previous paper, we developed a new move towards called obscure analogy that combines key aspects of logic based on ambiguous logic and similarity. However, obscure similarities can only be used if the possible values of the classified properties are derived from a numeric field. The aim of the present study is to broaden our previous approach to the systematic processing of confidential data. For this purpose, the obscure K-Mots algorithm is used using two boot technique. The presentation of the proposed approach is compared with classical analogue the International Software Benchmarking Standards Group (ISBSG) dataset.

Kaushik et al. (2015) introduce a new design method for estimating software costs using polynomial neural networks (PNNs) and intuitive ambiguity kits, resulting in improved SCE. Tests the presentation of a particular model on the many software development data available, especially the COCOMO81, NASA93 and Maxwell data sets. The specific method of using IFCM (intuitionistic fuzzy C Means) PMS is significantly better than PNN, which is significantly better than using PNN.

Sheta et al. (2010) Comparisons between particle swarm optimisation (PSO) algorithm, fuzzy logic (FL), and software cost assessment model are made using known cost opinion models such as Holsted, Walston Felix, Bailey-Basili, and Totti. Developed models are evaluated based on the NASA software magnitude of relative error (MMRE).

Parthasarathi et al. (Patra and Rajnish 2018) NASA used an ambiguous conditional algorithm to capture software project data and evaluate software efforts to create an appropriate model. We plan to create linear conditional models using the Potential Kilogram Domain (KLOC).

Attarzadeh and Ow (2009) The biggest challenge facing software developers in recent decades is predicting software organization development efforts based on development capabilities, size, complexity, and other dimensions. Project managers need to provide positive feedback on software development efforts. Most traditional technologies, such as operating points, weakening models, and COCOMO, require a lengthy evaluation process. New models of vague logic can create an alternative to this challenge. Combining ambiguous logic can solve many of the problems of existing effort evaluation models. This article describes an innovative fuzzy logic model for evaluating software maturity efforts and suggests a new approach to using vague logic in software effort ratings.

Ezghari and Zahi (2018) have studied the software Fuzzy Analogy based Software Effort Estimation model (FASEE) based on hopeless ambiguous similarities, successfully uses obscure logic with probabilistic logic to deal with the accuracy and rationality of conditions of uncertainty. Additionally, FASEE uses potential allocations to assess evaluation uncertainties, enabling software manager to assess risk. However, FASEE suffers from low data quality and doubt due to logical processing. In this study, we suggest the development of FASEE by establishing general criteria for overcoming the above shortcomings. Therefore, the basic model based on the standard obscure analogue known as Software Strength Assessment (CFASEE) has two possibilities. Displays the first range of attribute representations with a set of matte textures to match the effort of each attribute.

**5.2.1.2 K-nearest neighbour (KNN)** Huang et al. (2017) a novel developed a KNN imputation technology based on an incomplete event used by a cross-checking program to improve the parameters of each missing value. Test evaluation is done in eight grade databases under different circumstances. This study compares the specific stimulus approach to the median stimulus and the other three approaches to the KNN stimulus.

Idri et al. (2016) it has been found that the use of KNN stimulants instead of tolerance or exclusion may improve the predictive accuracy of analog methods. However, tolerance, elimination, and KNN computation affect the lost algorithm and MD percentage, both of which strongly adversely affect the accuracy of the attempted prediction.

Idri et al. (2018) investigate whether SVR improves the prognostic presentation of these two analog-based technologies when calculating missing data (MD) to replace KNN. 1134 tests were performed on seven databases: SVR / KNN MD from 10 to 90%.

Abnane et al. (2019) ASEE Group offers and evaluates KNN stimulant techniques. They compare the performance of ASEE with KNN imputation without a KNN imputation, or based on a group search, and without a parameter optimization with the KNN stimulus.

Kamei et al. (2008) suggest a new way to create artificial steps for the project and incorporate them into the relevant database to improve the evaluation efficiency of the software efforts provided by the analog program.

Satapathy and Rath (2017) Web applications using IFP-PUG use various machine building technologies such as K-Nearest Neighbour (KNN), Controlled Topographic Mapping (CTM),ivariate Adaptive Regression Splines (MARS), and arrangement and Regression Tree (CART). Function point approach.

Chinthanet et al. (2016) One of the long-standing discussions in ABE research is whether to re-evaluate past projects

(k values) when APE processes are based on the reuse of similar past projects.

**5.2.1.3 Artificial neural networks (ANN)** Saeed et al. (2018) a brief overview of past performance of software performance evaluation. The 10 studies that took part in the survey briefly explained how they help to solve the problem of power assessment according to time, cost or trial period. It states that different effort evaluations have advantages and disadvantages and that their application in different contexts is based on different types of historical data.

Whigham et al. (2015) offer the automatically transformed linear model (ATLM) as an ideal model for contrast with SEE methods. ATLM is simple but works well on a variety of projects. Additionally, ATMLM can be used with mixed measurements and classified data, and no stricture adjustment is required. This is crucial, i.e. it is ideal for copying the results obtained. These and other arguments for using the basic ATM model are presented and references are available and available. We recommend using ATLM efforts in the SEE as a basis for predicting the quality of future model comparisons.

Benala and Bandarupalli (2016) various methods have been suggested to get better the accurateness of ABE by adjust the obtained result. All published calibration methods depend on linear correction methods, except for the linear configuration based on the artificial neural network.

Bardsiri et al. (2012) have the quality of the tuition at ANN and the relevance of historical data were examined using the framework suggested in ABE. Two large and real data sets are used to evaluate the presentation of the planned method, and the results obtain are compare with the other eight methods.

Nassif et al. (2012) proposed a the Novel Artificial Neural Network (ANN) is used to predict software attempts to use the Use Case Point (UCP) model from used case maps. The data of this model are software size, productivity and complexity, and output output is software prediction. Introduced three independent variables (like ANN) and linear regression models with one dependent variable (attempt). Our database contains 240 databases, including 214 industrial and 26 educational projects.

Idri et al. (2016) Attempts to develop a program based on two analogs apply the missing data (MD) technology to the estimation technique: classical analog and obscure analog. Extra purposely, we analyze the prognostic routine of these two analog-based technologies with tolerance, elimination, or k-nearest neighbours (KNN) stimulus technique. Performed 1512 trials with seven data sets, three MD technique (tolerance, delete, KNN calculation), three absent algorithms (MAR: absent at random, NIM: non-ignorable missing) MD percentage range from 10 to 90%.

Kumari and Pushkar (2018) offer a new approach based on a crane search algorithm to predict software development efforts. It uses crane search to find the best features for the COCOMOII replica, and then further hybridizes to ANN to augment software processing correctness for better forecasting. Specific hybrid models are being tested on two standard data sets. During testing, the planned hybrid model give more precise and efficient results than other alive model.

**5.2.1.4 Convolutional neural network (CNN)** Abulalqader and Ali (2018). Learn about program cost reduction policies and how these strategies apply to general sections of the program. We offer basic algorithms in artificial astuteness, synthetic neural networks, genetic algorithm and fuzzy logic algorithm to decide which algorithm is most suitable for maximizing based on the best results seen in neural networks (FFNN, CNN, ENN, RBFN and NARX).

Ponalagusamy and Senthilkumar (2011) CNN tested the use of time multiplexing schemes to process large imagery using small sequences. Using the new addition algorithm, classification of inserted techniques, including RK techniques based on arithmetic mean (AM) and heronial means (HM).

Madari et al. (2019) have planned suggests a method to extend the function to increase accuracy. In this way the quadratic mapping feature is used for propagation. Better results can be achieved as dual cartography creates more unique features. Although application reflection is due to dimensional growth, the results are more accurate; especially when using WKNN (Weighted-K nearest neighbourhood) as regression model.

Darko et al. (2020) have studied Introduces the first comprehensive scientific study evaluating the latest AI-in-AECI study. Scientific mapping method was used for systematic and quantitative analysis of 41,827 relevant bibliographic documents obtained from Scopus.

Iqbal and Qureshi (2020) have studied multiple in-depth learning model used to construct text. In the in-depth learning process we will summarize the different models and give a complete understanding of the past, present and future of the text generation. In addition, the study included DL approaches studied and evaluated across NLP's various application domains.

Medrano and Aznarte (2020) have studied Spatio-temporal transport forecasting focuses on developing a complex in-depth neurological structure with relatively good performance, and demonstrates that it is compatible with multiple spatio-temporal conditions.

Camacho et al. (020) proposed a main contributions to this work are threefold: (1) we provide modern level social network analysis (SNA) with a periodical literature review; (2) we offer a new set of dimensions based on the four important features (or dimensions) of SNA; (3) Finally, we

present quantitative analysis of trendy SNA tools and structures. We also conducted scientific measurement research to identify the most active research sites and request domain in the region. This paper propose the meaning of four different dimensions used to define new dimensions (so-called degrees) for evaluating various software tools and structures, knowledge discovery, data integration and integration, measurement and visualization. SNA (20 analyzes of previous SNS and a standard set of SNA software tools).

**5.2.1.5 Deep neural network (DNN)** The focus is on defining big data; Architectural sites that support data analysis; Continue to use the analytical techniques mentioned above on complex bio communication issues. The challenges and future possibilities of big data analysis in bio communication are briefly discussed. In Nagaraj et al. (2018), complete summary of several data analysis techniques used by bioinformatics researchers and computer science.

Bosu et al. (2018) have studied The DNN and RetNet SEE models were used for two SEE databases aimed at solving the problem of infinite uncertainty and evaluating Renet's relevance as a potential SEE benchmark model. The Tisanes database was unstable as both model types achieved decision stability in the Kitchen ham database. Elastic Net works best with DNN, so it is not optional to use the SEE benchmark model.

Michoski et al. (2020) Proves that DNN competes with the freedom required for certain accuracy. In addition, the DNN-based approach has been expanded to include performance improvement and simultaneous parameter space exploration. Subsequently, the compressive magneto hydrodynamics (MHD) shock is resolved, and the situation where test data are used to upgrade the PDE system is not sufficient to confirm the observed / experimental data. This is achieved by enriching the sample PDE system with keywords and then managing the synthetic test data using the Supervision manual.

Mehta et al. (2019) proposed a Getting acquainted (gain, gain) with today's technologies that came from physics is easy to understand and intuitive for physicists. The review includes group models, in-depth learning, neural networks, clustering, data visualization, energy-based models, and variation methods.

Zarei and Asl (2020) have studied the continuous attribute choice method is used to select the most useful description. Finally, the features selected for the classification of OSA patients and general subjects are presented in different categories. Physiotherapy apnoea-ECG and fantasy data sets are used to evaluate the OSA detection method and ETR extraction algorithms, respectively. The Gentle Boost Classifier holds the record with 93.26% and 100% accuracy in one category, respectively. The proposed OSA automation system is better than all other modern methods.

Lu et al. (2020) have studied a discuss these different methods of visualization and consider the pros and cons of achieving a hyper-spectral image at industry-related speeds. When evaluating assignments in the spectral and spatial fields, the various methods of data processing / analysis and related steps, from pre-processing of data to the creation of an actual model, are discussed later.

Reynolds et al. (2018) the dynamic modelling methods of the building are discussed as the buildings are to be careful as active and supple partners in the district electrical system. In both cases, special attention is paid to models based on artificial intelligence, which are ideal for optimizing the direct performance of multiple vector system. Future research instructions given in this reconsider should include energy conversion departments, power grids, dynamic model development, and simple energy conservation models for district optimization.

### 6 Comparative analysis of state-of-art machine learning techniques in ABE

In this part, we compare the dissimilar parameters used in the technique that are approved out base on the classification obtainable in preceding piece. Investigating different papers and considering their use of parameters we give a cross mark for the respective reference papers regarding the techniques used, such supervised and unsupervised. We consider the following parameters for performance analysis of different research papers are accuracy (A), precision (P), recall (R), F-measure (F), sensitivity (SE), specificity (SP), mean magnitude of relative error (MMRE), median of the magnitude of relative error (MdmRE), root mean square error (RMSE), and mean absolute error (MAE). We elaborate them and a graphical representation is presented regarding the percentages of the parameters used in the reference papers. However, the performance analysis make the simplification of marks more justified since the research paper is evaluate using different type of machine learning techniques. The detailed description is carried in following section.

The main issues of the ABE structure are parallel occupation, resolution occupation and KNN. The various

machine learning techniques used in ABE are support vector machines (SVM), Naïve Bayes, Nearest neighbours, decision trees, neural networks, fuzzy logic, k-nearest neighbour (KNN), artificial neural network (ANN), convolutional neural network (CNN) and Deep neural network(DNN). In Table 1 the strength and weakness of machine learning techniques for analogy-based software effort estimation is summarised. The strengths of the SVM are (1) estimates by consulting experts (2) it is more effective high dimensional spaces. The weakness of SVM is that it does not perform well when the data has more noise i.e. target classes. Naïve Bayes is less sensitive to missing data and the algorithm is also relatively simple. Naïve Bayes is very sensitive to the form of input data. The strengths of the nearest neighbour are (1) it can be used for both classification and regression. (2) It does not explicitly build any model. Naïve Bayes works only on small number of input variables. It has no capability to dealing missing value problem. Decision tress does not require both data normalisation and data scaling. The weakness of decision tree is (1) it requires more memory. (2) space and time complexity are relatively higher. Neural networks have the ability to learn by themselves and produce the output and it performs multiple tasks in parallel without affecting the performance. Little bit over-hyped at the moment and the expectations exceed what can really be done is the weakness of the neural networks. Fuzzy logic is a dynamic supportive network and it is utilized in NL with different applications in artificial intelligence.

The weakness of fuzzy logic is (1) it can't perceive AI just as neural system type design (2) Enrolment capacities are a troublesome undertaking. KNN stores training dataset and learns from time of making real predictions and it requires no training before making prediction. The weakness of KNN is (1) it does not work well with high dimensions (2) it is sensitive to noise data, missing values and outliers. ANN has the ability to work with inadequate knowledge and ability to train machine. The weakness of ANN is (1) it has the difficulty of showing the problem to the network (2) assurance of proper network structure. CNN contains data requirements leading to over fitting and under fitting. CNN has high computational cost. DNN has the ability to work with incomplete knowledge and the traditional programming is stored on the

**Table 1** Quality assessment questions

No	Questions
QA1	Does the planned method get better the presentation of ABE?
QA2	Are the objectives of the study obviously distinct?
QA3	Is the specific evaluation method compare to other methods?
QA4	Are the limits of study analyze explicitly?
QA5	Are survey results available to support the findings of the paper?
QA6	Is this research improving the educational or professional community?
QA7	Is evaluation accuracy measured and reported?

**Table 2** Summary of machine learning techniques for analogy based software effort estimation

Machine learning classes	Machine learning technique	Strength towards ABE	Weakness towards ABE
Supervised learning	SVM	Estimation by consulting experts More effective in high dimensional spaces Relatively memory efficient	Not suitable for large data sets Does not perform very well when the data has more noise i.e. target classes
	Navie Bayes	Higher speed for large numbers of training and queries Less sensitive to missing data, the algorithm is also relatively simple	Very sensitive to the form of input data Assumption of independent predictor features
	Nearest neighbors	Can be used both for Classification and Regression Does not explicitly build any model	It works well on small number of input variables only No capability to dealing missing value problem
	Decision tree	Not required data normalization Not required data scaling	Require more memory Space and time complexity is relatively higher
	Neural networks	Ability to learn by themselves and produce the output Perform multiple tasks in parallel without affecting the performance	Little bit over-hyped at the moment and the expectations exceed what can be really done
	Unsupervised learning	Fuzzy logic	A dynamic emotionally supportive networks It utilized in NL with different applications in artificial intelligence
KNN		It stores training dataset and learns from time of making real predictions It requires no training before making prediction	Does not work well with high dimensions Sensitive to noisy data, missing values and outliers
ANN		The ability to work with inadequate knowledge Ability to train machine Parallel processing ability	The difficulty of showing the problem to the network Assurance of proper network structure
CNN		Data requirements leading to over fitting and under fitting Parameter-to-memory requirements	Don't have a good GPU they are quite slow to train High computational cost
DNN		Ability to work with incomplete knowledge Parallel processing capability Traditional programming is stored on the entire network	Unexplained behavior of the network Difficulty of showing the problem to the network

**Table 3** Performance comparison of different SVM techniques towards the analogy based software effort estimation

Ref. no	Performance metrics (%)								
	A	P	SE	S	FM	R	MMRE	RMSE	MBRE
Gupta et al. (Sikka and Verma 2011)	x			x			x		
Shashank et al. (Satapathy and Rath 2014)	x	x					x	x	
Tirimula et al. (Benala and Bandarupalli 2016)	x	x					x		x
Lin et al. (Lin et al. 2011)	x						x		
Pospieszny et al. (Czarnacka-Chrobot and Kobylinski 2018)	x		x	x			x	x	x
Oliveira et al. (2010)	x						x		
Corazza et al. (2011)	x		x				x		

entire network. However, DNN is unexplained behaviour of network and it has the difficulty in showing the problem to the network. Table 2 summarizes the strength and weakness of different machine learning techniques in ABE.

### 6.1 Performance comparison of SVM classifier for software effort estimation

Table 3 presents the relative analysis of SVM analogy based software effort estimation. The table clearly shows the accuracy of SVM in Gupta et al. (2011) gives 75%, MMRE of 25.72. In Satapathy and Rath (2014) gives the RMSE of 0.145, Benala and Bandarupalli (2016) MBRE of 0.810 and Pospieszny et al. (2018) has both RMSE, MBRE of 0.27 and 0.16 respectively. Therefore, the accuracy in Oliveira et al. (2010) is much greater compare to other state-of-art SVM techniques but the MMRE value is less in this paper. However, the MMRE value in Corazza et al. (2011) is greater than other SVM techniques. The graphical representation of different SVM based ABE techniques are summarized in Fig. 4. The plot clearly depicts the comparative analysis of different SVM techniques in terms of accuracy and MMRE.

### 6.2 Performance comparison of Naive Byes classifier for software effort estimation

Table 4 summarizes the comparative analysis of Naive Byes analogy based software effort estimation. The table clearly shows the Naive Byes value in Braga et al. (2007) gives the RMSE of 0.3031, MMRE of 0.0803 and MAE of 0.2647. In Zhang et al. (2017) gives the SE of 90% and SP of 61.1%. In BaniMustafa (2018) and Hussain et al. (2013) has Precision of 88% and 0.786, Recall of 93% and 0.688 respectively. There is an addition parameter in Hussain et al. (2013) is F-measure of 0.733. However, precision and recall in BaniMustafa (2018) is much greater compare to other NB techniques. The graphical representation of different Naive Byes based ABE techniques are summarized in Fig. 5. The plot clearly depicts the comparative analysis of different Naive Byes techniques in terms of precession and Recall. Pospieszny et al.

Fig. 4 Comparative analysis of SVM based ABE

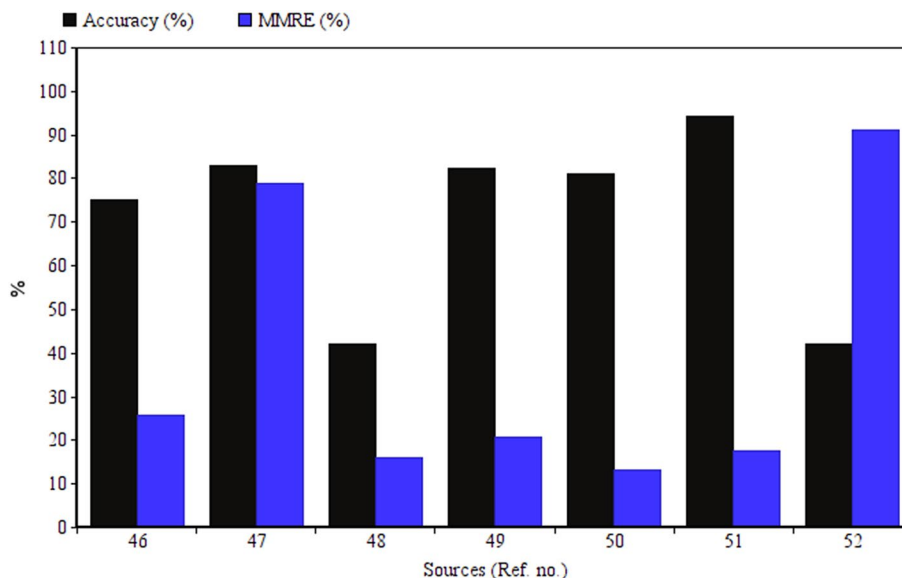
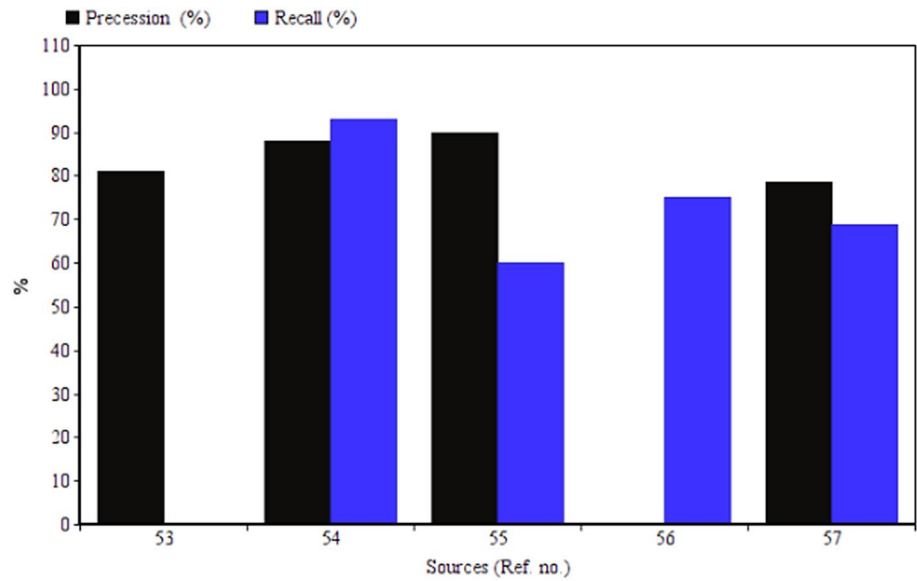


Table 4 Performance comparison of different Neive bytes techniques towards the analogy based software effort estimation

Ref. no	Performance metrics (%)								
	A	P	SE	SP	FM	R	MMRE	RMSE	MAE
Shivhare et al. (Braga, Oliveira and Meira 2007)							x	x	x
BaniMustafa (2018)		x				x			
Zhang et al. (2017)			x	x					
Wen et al. (Zhang et al. 2015)		x					x		
Hussain et al. (2013)		x			x	x			

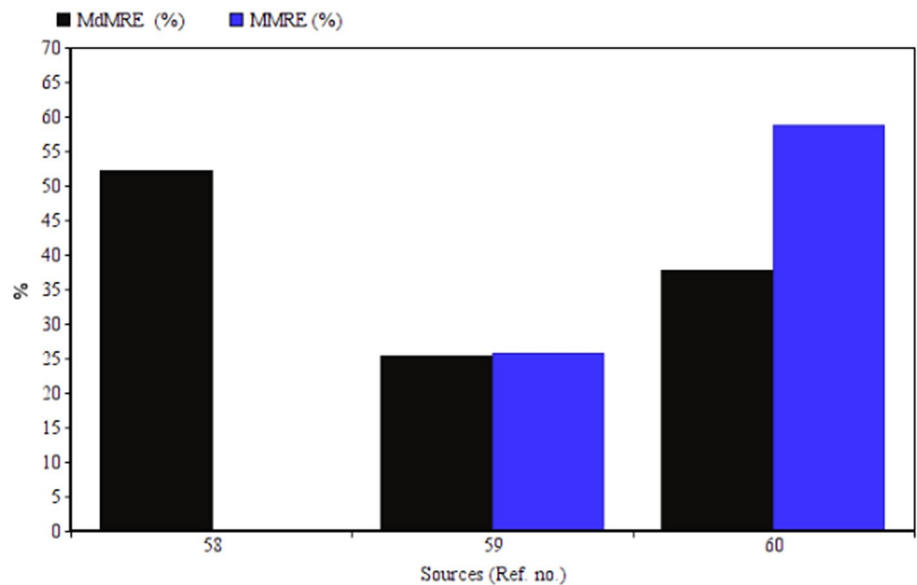
**Fig. 5** Comparative analysis of Naive Byes based ABE



**Table 5** Performance comparison of different Nearest-Neighbor techniques towards the analogy based software effort estimation

Ref. no	Performance metrics (%)									
	A	P	SE	S	FM	R	MMRE	MdMRE	MAE	MdAE
Sarro and Petrozziello (2018)				x					x	x
Li et al. (2008)	x				x		x	x		
Le-Do et al. (2010)	x						x	x		

**Fig. 6** Comparative analysis of Naive Byes based ABE



**6.3 Performance comparison of nearest-neighbour classifier for software effort estimation**

Table 5 gives the comparative analysis of nearest-neighbour techniques of analogy based software effort estimation. The table clearly shows the nearest-neighbour value in Sarro and

Petrozziello (2018) gives the MAE of 0.478 and MdAE of 0.523. In Li et al. (2008) and Le-Do et al. (2010) has Accuracy of 25% and 32.47%, MMRE of 25% and 58.71% and MdmRE of 25% and 37.72% respectively. Therefore Accuracy, MMRE and MdmRE in Le-Do et al. (2010) are much greater compare to other nearest-neighbour techniques. The



graphical representation of the different nearest-neighbour based ABE techniques are summarized in Fig. 6. The plot clearly depicts the comparative analysis of different nearest-neighbour techniques in terms of MdMRE and MMRE.

### 6.4 Performance comparison of decision tree classifier for software effort estimation

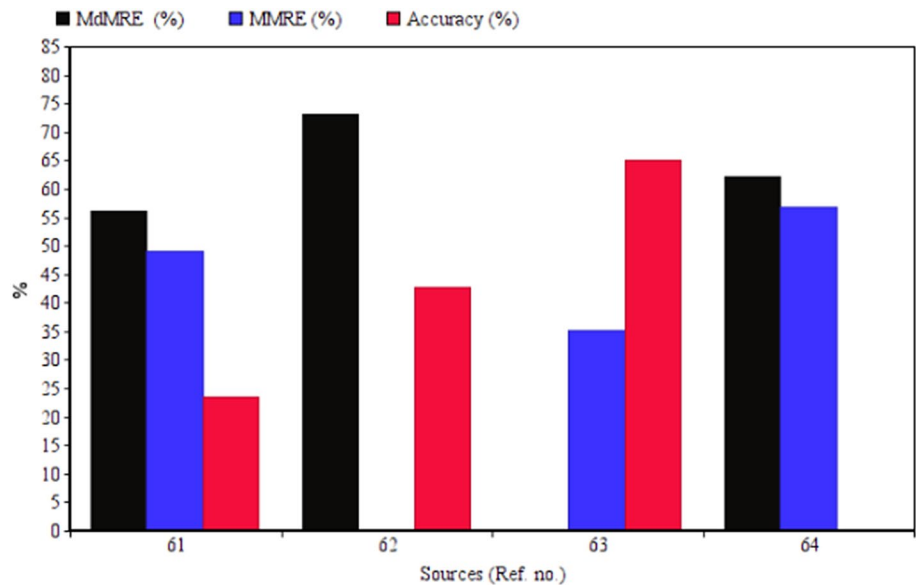
Table 6 gives the comparative analysis of decision tree analogy based software effort estimation. The table clearly shows the decision tree value in Nassif et al. (2013) gives the accuracy of 23%, MMRE of 0.49 and MdMRE of 0.56.

In Mohanty et al. (2010), Papatheocharous and Andreou (2009) has Accuracy of 42% and 65%. In Papatheocharous and Andreou (2009), Trendowicz and Jeffery (2014) has MMRE of 0.35 and 0.56 and in Mohanty et al. (2010), Trendowicz and Jeffery (2014) has MdMRE of 0.73% and 0.62%. Therefore Accuracy, MMRE and MdMRE in Nassif et al. (2013) are much lesser compare to other decision tree techniques. The graphical representation of the different decision tree based ABE techniques are summarized in Fig. 7. The plot clearly depicts the comparative analysis of different decision tree techniques in terms of accuracy, MdMRE and MMRE.

**Table 6** Performance comparison of different Decision Tree techniques towards the analogy based software effort estimation

Ref. no	Performance metrics (%)							
	A	P	SE	SP	FM	R	MMRE	MdMRE
Nassif et al. (2013)	x	x		x			x	x
Mohanty et al. (2010)	x		x		x			x
Papatheocharous and Andreou (2009)	x						x	
Trendowicz and Jeffery (2014)				x			x	x

**Fig. 7** Comparative analysis of decision tree based ABE



**Table 7** Performance comparison of different Neural Network techniques towards the analogy based software effort estimation

Ref. no	Performance metrics (%)								
	A	P	SE	SP	FM	R	MMRE	MdMRE	RMSE
Idri et al. (Zakrani and Zahi 2010)	x				x			x	
Kumar et al. (2008a, b)	x						x	x	
de Campos Souza et al. (2019)			x						x
Kaushik and Singal (2019)	x						x	x	
Huang and Chiu (2009)	x						x	x	
Rao and Kumar (2015)			x				x	x	

### 6.5 Performance comparison of neural network classifier for software effort estimation

Table 7 gives the comparative analysis of neural network techniques analogy based software effort estimation. The table uses parameters such as accuracy, MdmRE, MMRE and RMSE which clearly shows the neural network value. In Idri et al. (2010) gives the accuracy of 0.52% and MdmRE of 0.67. In de Campos Souza et al. (2019) gives the RMSE value of 145.13. In Kumar et al. (2008a, b), Kaushik and Singal (2019), Huang and Chiu (2009) has Accuracy of 0.416, 75.71 and 0.43%, MMRE of 0.579, 27.13 and 0.28 and MdmRE of 0.545, 29.69 and 0.27%. However, the accuracy, MMRE and MdmRE in Kumar et al. (2008a, b) are much lesser compare to other neural network techniques (Leung 2002). The graphical representation of the different neural network based ABE techniques are summarized in Fig. 8. The plot clearly depicts the comparative analysis of different neural network techniques in terms of accuracy, MdmRE and MMRE.

### 6.6 Performance comparison of fuzzy logic classifier for software effort estimation

Table 8 shows the comparative analysis of fuzzy analogy-based software effort estimation. The table clearly shows the accuracy in Azzeh et al. (2008) of 44%, MMRE (41%) and MdmRE (31.6%). In Amazal et al. (2014a, b) gives an accuracy of (46%) and MMRE (51%). In Kaushik et al. (2015) gives accuracy (0.80), MMRE (19.80), RMSE (7.69) and Recall (5). In Kaushik et al. (2015) gives a superior number of parameters whereas the values are lesser as compared to others. Thus, the greater values were seen in Kaushik et al. (2015). The graphical representation of the different fuzzy based ABE techniques are summarized in Fig. 9. The plot clearly depicts the comparative analysis of different fuzzy techniques in terms of accuracy, MdmRE and MMRE.

Fig. 8 Comparative analysis of neural network based ABE

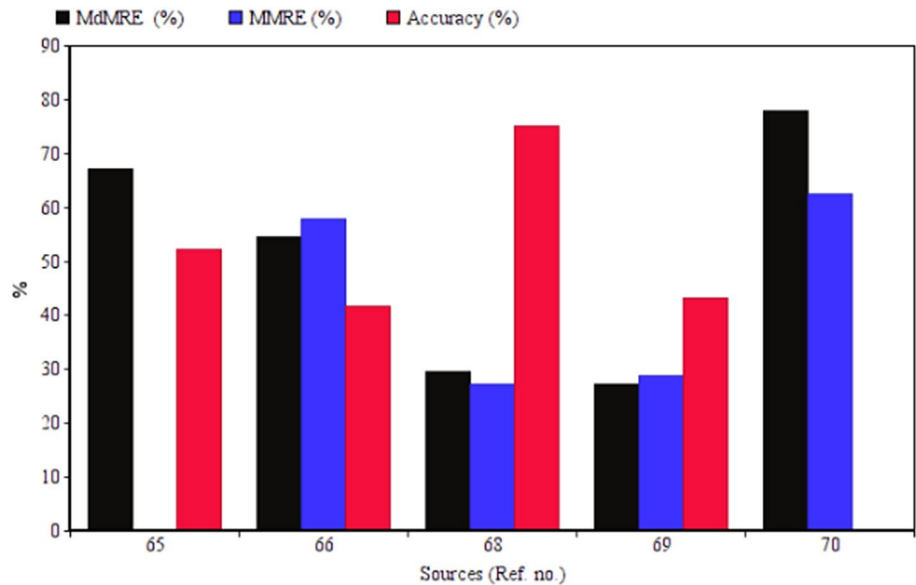
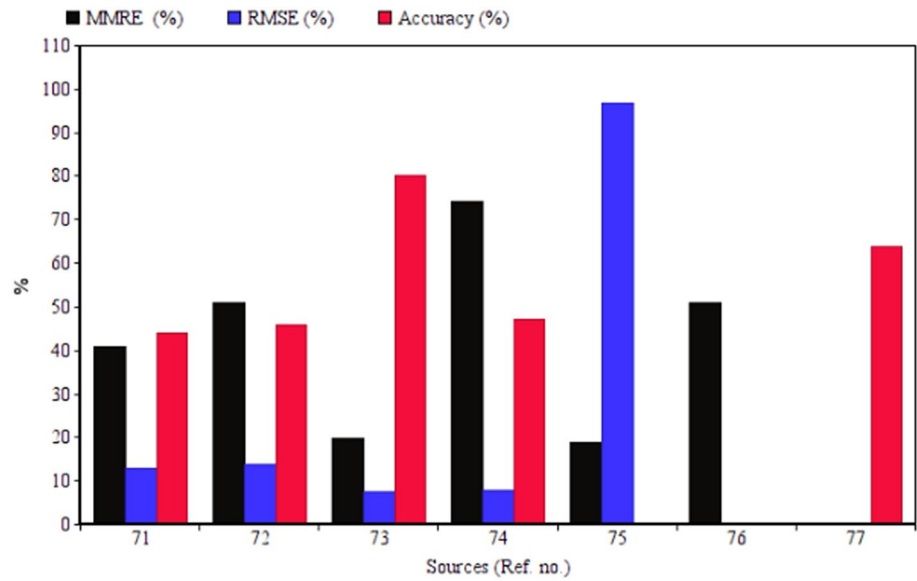


Table 8 Performance comparison of different FUZZY techniques towards the analogy-based software effort estimation

Ref. no	Performance metrics (%)										
	A	P	SE	S	FM	R	MMRE	RMSE	MPRE	MRE	MdmRE
Azzeh et al. (2008)	x						x			x	x
Amazal et al. (2014a, b)	x	x					x				
Kaushik et al. (2015)	x					x	x	x			
Sheta et al. (2010)							x			x	
Parthasarathi et al. (Patra and Rajnish 2018)			x				x	x			
Attarzadeh and Ow (2009)							x		x		x
(Ezghari and Zahi 2018)	x	x									

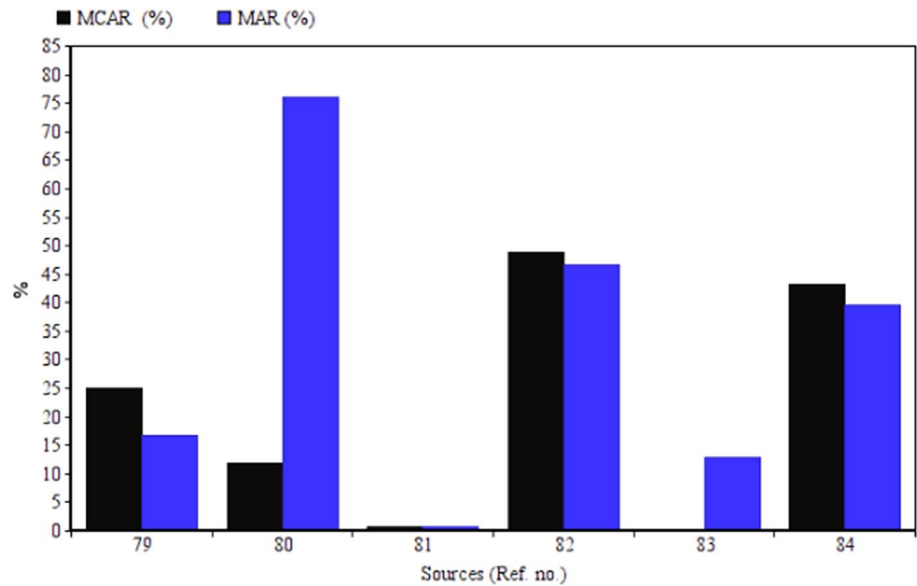
**Fig. 9** Comparative analysis of Fuzzy logic based ABE



**Table 9** Performance comparison of different KNN techniques towards the analogy-based software effort estimation

Ref. no	Performance metrics (%)										
	A	P	SE	S	FM	R	MMRE	MCAR	MAR	NIM	MEI
Huang et al. (2017)	x										x
Idri et al. (2016)		x						x	x	x	
Idri et al. (2018)					x			x	x	x	
Abnane et al. (2019)								x	x	x	
Kamei et al. (2008)	x						x				
Satapathy and Rath (2017)	x						x				
Chinthanet et al. (2016)							x				

**Fig. 10** Comparative analysis of KNN based ABE



**Table 10** Performance comparison of different ANN techniques towards the analogy-based software effort estimation

Ref. no	Performance metrics (%)										
	A	P	SE	S	FM	R	MMRE	RMSE	RRSE	MAE	RAE
Saeed et al. (2018)							x	x	x	x	x
Whigham et al. (2015)	x						x				
Benala and Bandrupalli (2016)	x						x				
Bardsiri et al. (2012)	x						x			x	
Nassif et al. (2012)	x						x				
Idri et al. (2016)		x						x			x
Kumari and Pushkar (2018)	x						x				

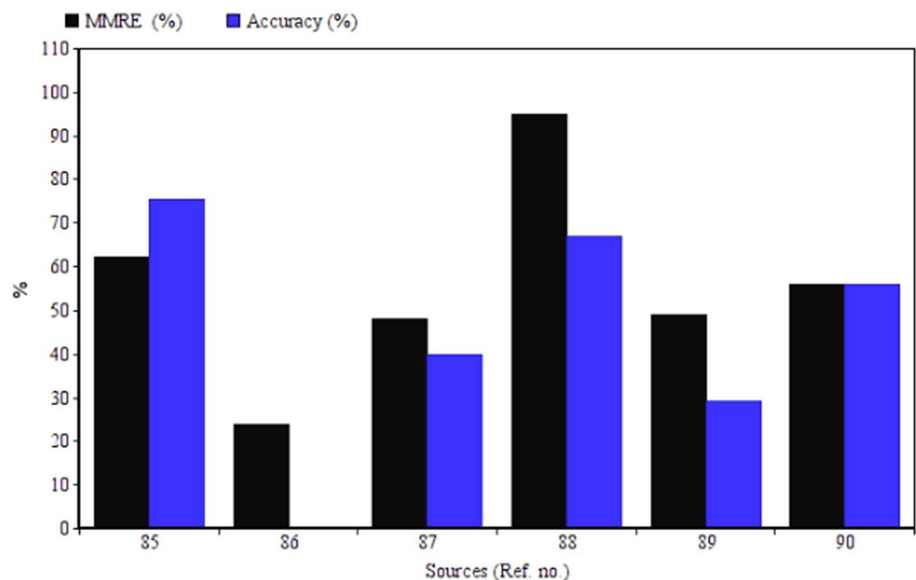
**6.7 Performance comparison of KNN classifier for software effort estimation**

Table 9 shows the comparative analysis of KNN analogy-based software effort estimation. The table clearly shows the values of MCAR (0.025), MAR (0.0167) and NIM (0.025) in Idri et al. (2016). In Idri et al. (2018) it gives MCAR (0.012), MAR (0.0076) and NIM (0.0076). The additional parameters in KNN are MCAR, MAR and NIM. The values in Idri et al. (2016) consist of greater values. In Kamei et al. (2008) it gives values of accuracy (0.478) and MMRE (0.488) whereas in Kamei et al. (2008) it gives values of accuracy (39.1) and MMRE (6.904). Thus, the greater values were seen in Kamei et al. (2008). The graphical representation of the different KNN based ABE techniques are summarized in Fig. 10. The plot clearly depicts the comparative analysis of different KNN techniques in terms of MCAR and MMRE.

**6.8 Performance comparison of ANN classifier for software effort estimation**

Table 10 shows the comparative analysis of ANN analogy-based software effort estimation. The additional parameters are RMSE, RAE, RRSE and MAE. In Saeed et al. (2018) it gives the values of MMRE (6.21), RMSE (0.065), RAE (9.71), RRSE (17.81) and MAE (0.026). The table clearly shows the accuracy in Whigham et al. (2015) of 0.43 and MMRE (2.79). In Benala and Bandrupalli (2016) gives an accuracy of (0.40) and MMRE (4.7974). In Bardsiri et al. (2012) gives accuracy (00.67) and MMRE (0.95). In Saeed et al. (2018) gives a superior number of parameters whereas the values are lesser as compared to others. Thus, the greater values were seen in Benala and Bandrupalli (2016). The graphical representation of the different ANN based ABE techniques are summarized in Fig. 11. The plot clearly depicts the comparative analysis of different ANN techniques in terms of accuracy and MMRE.

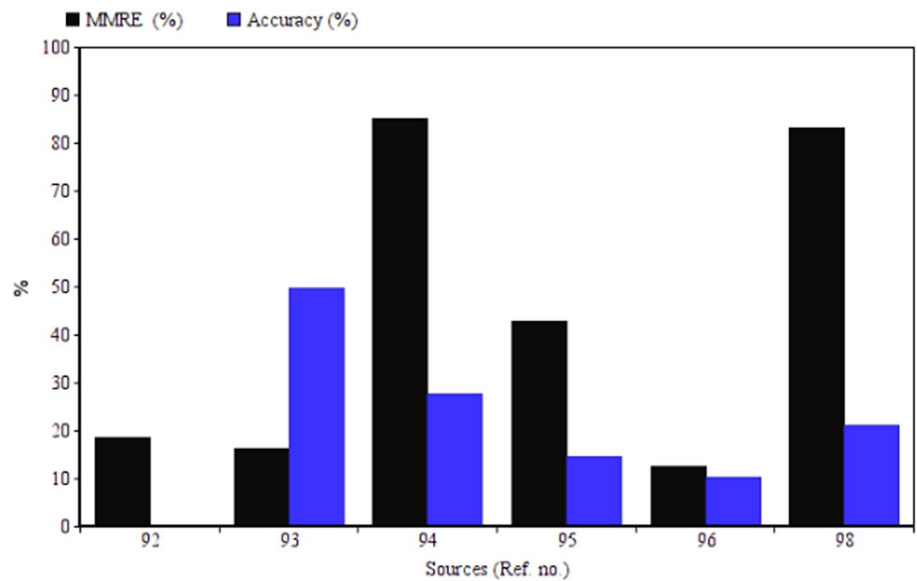
**Fig. 11** Comparative analysis of ANN based ABE



**Table 11** Performance comparison of different CNN techniques towards the analogy-based software effort estimation

Ref. no	Performance metrics (%)											
	A	P	SE	S	FM	R	MMRE	RMSE	RRSE	BRE	VAF	
Abulalqader and (Ali 2018)							x	x		x	x	
Ponalagusamy and Senthilkumar (2011)	x						x					
Madari et al. (2019)	x						x					
Darko et al. (2020)	x						x			x		
Iqbal and Qureshi (2020)	x						x					
Medrano and Aznarte (2020)		x						x				
Camacho et al. (2020)	x						x					

**Fig. 12** Comparative analysis of CNN based ABE



**Table 12** Performance comparison of different DNN techniques towards the analogy-based software effort estimation

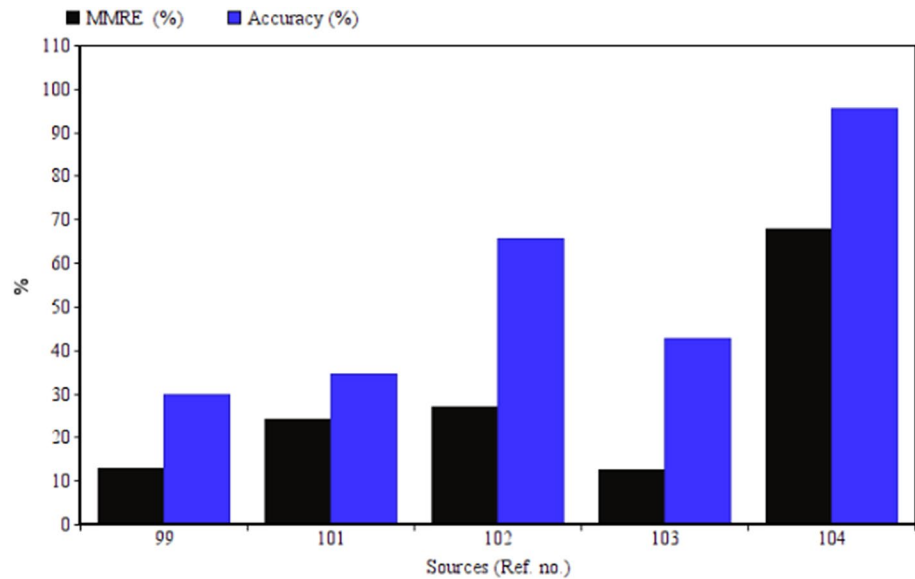
Ref. no	Performance metrics (%)											
	A	P	SE	S	FM	R	MMRE	RMSE	MSE	MAE	LSD	
Nagaraj et al. (2018)										x	x	
Bosu et al. (2018)	x						x			x	x	
Michoski et al. (2020)	x						x					
Mehta et al. (2019)	x	x					x			x		
Zarei and Asl (2020)	x						x					
Lu et al. (2020)	x						x					
Reynolds et al. (2018)								x	x	x		

**6.9 Performance comparison of CNN classifier for software effort estimation**

Table 11 shows the comparative analysis of CNN analogy-based software effort estimation. The additional parameters are RMSE, BRE and VAF. In Abulalqader and Ali (2018). It gives the values of MMRE (1.8460), RMSE (71.3928), BRE (0.2501), and VAF (93.6742). The table clearly shows the accuracy in Ponalagusamy and Senthilkumar (2011)

of 0.497 and MMRE (1.642). In Madari et al. (2019) gives an accuracy of (27.8) and MMRE (85.26). In Darko et al. (2020) gives accuracy (1.469) and MMRE (4.28). In Abulalqader and Ali (2018), they give a superior number of parameters whereas the values. Thus, the greater values were seen in Madari et al. (2019). The graphical representation of the different CNN based ABE techniques are summarized in Fig. 12. The plot clearly depicts the comparative analysis of different CNN techniques in terms of accuracy and MMRE.

**Fig. 13** Comparative analysis of DNN based ABE



### 6.10 Performance comparison of DNN classifier for software effort estimation

Table 12 shows the comparative analysis of DNN analogy-based software effort estimation. The additional parameters are RMSE, MSE and MAE. In Reynolds et al. (2018) it gives the values of RMSE (0.6408), MSE (0.0016) and MAE (0.0399). The table clearly shows the accuracy in Michoski et al. (2020) of 0.3447 and MMRE (2.428). In Mehta et al. (2019) gives an accuracy of (0.6556) and MMRE (0.2696). In Mehta et al. (2019) gives accuracy (0.428) and MMRE (1.244). In Reynolds et al. (2018) it gives a superior number of parameters whereas the values are lesser as compared to others. Thus, the greater values were seen in Mehta et al. (2019). The graphical representation of the different DNN based ABE techniques are summarized in Fig. 13. The plot clearly depicts the comparative analysis of different DNN techniques in terms of accuracy and MMRE.

## 7 Conclusion

This study considers the development of ABE as a generally conventional evaluation methodology in the field of software progress organization evaluation. Five key question were identified to explain the direction of the explore. Finally, the results of the current study are shortened as follow:

- *RQ1* How did ABE improve in prior studies?

Certain study be classified base on two groups: supervised and unsupervised. The first group is divided into two divisions as classification based technique and regression based technique. Classification based technique are

divided into three divisions: SVM, Naive Bayes and Nearest Neighbours. The regression-based technique is divided into two divisions: Decision Tree and the neural network. The second group is divided into 5 divisions: Fuzzy, KNN, ANN, CNN and DNN. Each group is fully explained and their strengths and weaknesses are listed accordingly. Actually, the above mentioned group are the answer of this explore enquiry.

- *RQ2* How is the accuracy of the models recommended in prior studies calculated?

As the current study confirms, the most selective performance measures (90% and 94%, respectively) are MMRE and A. Although these measurements have been widely used in previous studies, arithmetical tests should also be used to let alone inclined and unequal results. This is due to the modern high level of criticism of MRE-based measurements.

- *RQ3* How were the machine learning (ML) techniques is compared to the other models in previous studies?

Machine learning confirms the lack of experiential research on the use of BN, SVR, GA, GP, and AR technologies. a number of ML devices are not used in the SDEE field. However, researchers are urged to do extra experimental research on the hardly ever used ML technology, which extra strengthens the experiential proof for its effectiveness.

- *RQ4* Did we get the same attention from researchers at different stages of the analogy process?

The classification of the different stages of the analogy process attracts the attention of researchers.

- *RQ5*: Which mechanisms of ABE (similarity and solution functions) were used in previous studies?

The ABE similarity function is used to determine the similarity by comparing the characteristics of the two circuits. There are two admired function of unity: the Euclidean similarity (ES) and the Manhattan similarity (MS). The MS method is alike to ES, but calculate the completed is similarity among the attribute. There are other similarities, such as average equivalence rank, maximum distance equivalence, and Minkowski equivalence used in previous studies. Solution features are used to evaluate software development efforts by identifying similar projects as similar processes. Features of popular solutions: similarity of similar projects, average of similar projects, average of similar projects, average distance weight average. K represents the average effort for similar projects.

Accordingly, five study questions were considered in the present study. The most important limitation of this study is the various criteria used to evaluate the efforts made in previous studies. This difference allows us to evaluate the correctness of the model recommended in this section. For example, in previous studies, performance measurements, evaluation methods, and data sets were completely different. As a future activity, we are improving the accuracy of ABE ratings for classified projects using sophisticated computer technology.

## References

- Abnane I, Hosni M, Idri A and Abran A (2019) Analogy software effort estimation using ensemble KNN imputation. In: 2019 45th Euromicro Conference on software engineering and advanced applications (SEAA). IEEE, pp 228–235
- Abulalqader FA, Ali AW (2018). Comparing Different Estimation Methods for Software Effort. In: 2018 1st Annual International Conference on information and sciences (AiCIS). IEEE, pp 13–22
- Ahmed MA, Muzaffar Z (2009) Handling imprecision and uncertainty in software development effort prediction: A type-2 fuzzy logic based framework. *Inf Softw Technol* 51(3):640–654
- Alpaydin E (2004) Introduction to machine learning. The MIT Press, Cambridge
- Amazal FA, Idri A, Abran A (2014a) An analogy-based approach to estimation of software development effort using categorical data. In: 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on software process and product measurement. IEEE, pp 252–262
- Amazal FA, Idri A, Abran A (2014b) Improving fuzzy analogy based software development effort estimation. In: 2014 21st Asia-Pacific Software Engineering Conference, 1: 247–254. IEEE.
- Angelis L, Stamelos I (2000) A simulation tool for efficient analogy-based cost estimation. *Empir Softw Eng* 5(1):35–68
- Attarzadeh I, Ow SH (2009) Software development effort estimation based on a new fuzzy logic model. *Int J Comput Theory Eng* 1(4):473
- Azzeh M, Neagu D, Cowling P (2008) Improving analogy software effort estimation using fuzzy feature subset selection algorithm. In: Proceedings of the 4th International Workshop on predictor models in software engineering, pp 71–78
- Azzeh M, Neagu D, Cowling P (2009) Software effort estimation based on weighted fuzzy grey relational analysis. In: Proceedings of the 5th International Conference on predictor models in software engineering, Canada
- Azzeh M, Neagu D, Cowling PI (2011) Analogy-based software effort estimation using Fuzzy numbers. *J Syst Softw* 84(2):270–284
- BaniMustafa A (2018) Predicting software effort estimation using machine learning techniques. In: 2018 8th International Conference on computer science and information technology (CSIT). IEEE, pp 249–256
- Bardsiri VK, Jawawi DNA, Hashim SZM, Khatibi E (2012) Increasing the accuracy of software development effort estimation using projects clustering. *IET Softw* 6(6):461–473
- Bardsiri VK, Abang Jawawi DN, Khatibi E (2014) Towards improvement of analogy-based software development effort estimation: a review. *Int J Softw Eng Knowl Eng* 24(07):1065–1089
- Benala TR, Bandarupalli R (2016) Least square support vector machine in analogy-based software development effort estimation. In: 2016 International Conference on recent advances and innovations in engineering (ICRAIE). IEEE, pp 1–6
- Bibi S, Stamelos I (2006) Selecting the appropriate machine learning techniques for the prediction of software development costs. In: Maglogiannis I, Karpouzis K, Bramer M (eds) Artificial intelligence applications and innovations, vol 204, pp 533–540
- Boehm B, Clark B, Horowitz E, Westland C, Madachy R, Selby R (1995) Cost models for future software life cycle processes: Cocomo 2.0. *Ann Softw Eng* 1(1):57–94
- Boehm B, Abts C, Chulani S (2000) Software development cost estimation approaches—A survey. *Ann Softw Eng* 10(1–4):177–205
- Bosu MF, Mensah S, Bennin K, Abuaiadah D (2018) Revisiting the conclusion instability issue in software effort estimation
- Braga PL, Oliveira AL, Meira SR (2007) Software effort estimation using machine learning techniques with robust confidence intervals. In: 7th International Conference on hybrid intelligent systems (HIS 2007). IEEE, pp 352–357
- Camacho D, Panizo-LLedot A, Bello-Organ G, Gonzalez-Pardo A, Cambria E (2020) The four dimensions of social network analysis: an overview of research methods, applications, and software tools. *Inf Fusion* 63:88–120
- Chinthanet B, Phannachitta P, Kamei Y, Leelaprute P, Rungsawang A, Ubayashi N, Matsumoto K (2016) A review and comparison of methods for determining the best analogies in analogy-based software effort estimation. In: Proceedings of the 31st Annual ACM Symposium on applied computing, pp 1554–1557
- Chiu NH, Huang SJ (2007) The adjusted analogy-based software effort estimation based on similarity distances. *J Syst Softw* 80(4):628–640
- Corazza A, Di Martino S, Ferrucci F, Gravino C, Mendes E (2011) Investigating the use of Support Vector Regression for web effort estimation. *Empir Softw Eng* 16(2):211–243
- Darko A, Chan AP, Adabre MA, Edwards DJ, Hosseini MR, Ameyaw EE (2020) Artificial intelligence in the AEC industry: Scientometric analysis and visualization of research activities. *Autom Construct* 112:103081
- de Campos Souza PV, Guimaraes AJ, Araujo VS, Rezende TS, Araujo VJS (2019) Incremental regularized data density-based clustering neural networks to aid in the construction of effort forecasting systems in software development. *Appl Intell* 49(9):3221–3234
- Dragicevic S, Celar S, Turic M (2017) Bayesian network model for task effort estimation in agile software development. *J Syst Softw* 127:109–119

- Elish MO (2009) Improved estimation of software project effort using multiple additive regression trees. *Expert Syst Appl* 36(7):10774–10778
- El Bajta M (2015) Analogy-based software development effort estimation in global software development. In: 2015 IEEE 10th International Conference on global software engineering workshops. IEEE, pp 51–54
- Ezghari S, Zahi A (2018) Uncertainty management in software effort estimation using a consistent fuzzy analogy-based method. *Appl Soft Comput* 67:540–557
- Faria P, Miranda E (2012) Expert judgment in software estimation during the bid phase of a project—an exploratory survey. In: Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2012 Joint Conference of the 22nd International Workshop on. IEEE, pp 126–131
- Grimstad S, Jørgensen M (2009) Preliminary study of sequence effects in judgment-based software development work-effort estimation. *IET Softw* 3(5):435–441
- Gupta S, Sikka G, Verma H (2011) Recent methods for software effort estimation by analogy. *ACM SIGSOFT Softw Eng Notes* 36(4):1–5
- Huang SJ, Chiu NH (2009) Applying fuzzy neural network to estimate software development effort. *Appl Intell* 30(2):73–83
- Huang SJ, Chiu NH, Chen LW (2008) Integration of the grey relational analysis with genetic algorithm for software effort estimation. *Eur J Oper Res* 188(3):898–909
- Huang J, Keung JW, Sarro F, Li YF, Yu YT, Chan WK, Sun H (2017) Cross-validation based K nearest neighbor imputation for software quality datasets: An empirical study. *J Syst Softw* 132:226–252
- Hussain I, Kosseim L, Ormandjieva O (2013) Approximation of COSMIC functional size to support early effort estimation in Agile. *Data Knowl Eng* 85:2–14
- Idri A, Zakrani A, Zahi A (2010) Design of radial basis function neural networks for software effort estimation. *IJCSI Int J Comput Sci Issues* 7(4):595–611
- Idri A, Abnane I, Abran A (2016) Missing data techniques in analogy-based software development effort estimation. *J Syst Softw* 117:595–611
- Idri A, Abnane I, Abran A (2018) Support vector regression-based imputation in analogy based software development effort estimation. *J Softw Evol Process* 30(12):2114
- Iqbal T, Qureshi S (2020) The survey: text generation models in deep learning. *J King Saud Univ-Comput Inf Sci*
- Johnson RA, Wichern DW (2007) Applied multivariate statistical analysis, 6th edn. Prentice Hall
- Jørgensen M (2004) A review of studies on expert estimation of software development effort. *J Syst Softw* 70(1):37–60
- Jørgensen M (2016) Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays. *J Syst Softw* 117:274–281
- Kadoda G (2000) Experiences using case-based reasoning to predict software project effort. In: International Conference on empirical assessment and evaluation in software engineering (Keele University, 2000)
- Kamei Y, Keung J, Monden A, Matsumoto KI (2008) An over-sampling method for analogy-based software effort estimation. In: Proceedings of the Second ACM-IEEE International Symposium on Empirical software engineering and measurement, pp 312–314
- Kaushik A, Singal N (2019) A hybrid model of wavelet neural network and metaheuristic algorithm for software development effort estimation. *Int J Inf Technol*, pp 1–10
- Kaushik A, Soni AK, Soni R (2015) A hybrid approach for software cost estimation using polynomial neural networks and intuitionistic fuzzy sets. *Int J Comput Appl Technol* 52(4):292–304
- Khatibi V and Jawawi DN (2011) Software cost estimation methods: a review I. *J Emerg Trends in Comput Inf Sci*
- Kumar KV, Ravi V, Carr M, Kiran NR (2008a) Software development cost estimation using wavelet neural networks. *J Syst Softw* 81:1853–1867
- Kumar KV, Ravi V, Carr M, Kiran NR (2008b) Software development cost estimation using wavelet neural networks. *J Syst Softw* 81(11):1853–1867
- Kumari S, Pushkar S (2018) Cuckoo search based hybrid models for improving the accuracy of software effort estimation. *Microsyst Technol* 24(12):4767–4774
- Lamersdorf A, Münch J, Torre AFV, Sánchez CR, Rombach D (2014) Estimating the effort overhead in global software development. arXiv preprint [arXiv:1401.2730](https://arxiv.org/abs/1401.2730)
- Le-Do TK, Yoon KA, Seo YS, Bae DH (2010). Filtering of inconsistent software project data for analogy-based effort estimation. In: 2010 IEEE 34th Annual Computer Software and Applications Conference. IEEE, pp 503–508
- Leung HKN (2002) Estimating maintenance effort by analogy. *Empir Softw Eng* 7(2):157–175
- Li JZ, Ruhe G (2008) Software effort estimation by analogy using attribute selection based on rough set analysis. *Int J Softw Eng Knowl Eng* 18(1):1–23
- Li J, Ruhe G, Al-Emran A, Richter MM (2007) A flexible method for software effort estimation by analogy. *Empir Softw Eng* 12(1):65–106
- Li YF, Xie M, Goh TN (2008) Bayesian inference approach for probabilistic analogy-based software maintenance effort estimation. In: 2008 14th IEEE Pacific Rim International Symposium on dependable computing. IEEE, pp 176–183
- Lin JC, Chang CT and Huang SY (2011) Research on software effort estimation combined with genetic algorithm and support vector regression. In *2011 International Symposium on Computer Science and Society*: 349–352.
- Lu Y, Saeys W, Kim M, Peng Y, Lu R (2020) Hyperspectral imaging technology for quality and safety evaluation of horticultural products: a review and celebration of the past 20-year progress. *Postharvest Biol Technol* 170:111318
- Madari MJ, Bahrami H, Niazi M (2019) Improve software effort estimation using weighted KNN in new feature space. *IJRDT* 11(4):421–427
- Mair C, Kadoda G, Lefley M, Phalp KC, Schofield M, Shepperd S (2000) Webster, An investigation of machine learning based prediction systems. *J Syst Softw* 53(1):23–29
- Medrano R, Aznarte JL (2020) A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction. *Appl Soft Comput* 96:106615
- Mehta P, Bukov M, Wang CH, Day AG, Richardson C, Fisher CK, Schwab DJ (2019) A high-bias, low-variance introduction to machine learning for physicists. *Phys Rep* 810:1–124
- Mendes E (2008) The use of bayesian networks for Web effort estimation: further investigation. In: Proceedings of the 8th International Conference on Web Engineering, pp 203–216
- Mendes E, Mosley N, Watson I (2002) A comparison of case-based reasoning approaches to Web hypermedia project cost estimation. In: Proceedings of the 11th International Conference on World Wide Web (WWW'02), pp 272–280
- Michoski C, Milosavljević M, Oliver T, Hatch DR (2020) Solving differential equations using deep neural networks. *Neurocomputing* 399:193–212
- Mohanty R, Ravi V, Patra MR (2010) The application of intelligent and soft-computing techniques to software engineering problems: a review. *Int J Inf Decis Sci* 2(3):233–272



- Moosavi SHS, Bardsiri VK (2017) Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation. *Eng Appl Artif Intell* 60:1–15
- Nagaraj K, Sharvani GS, Sridhar A (2018) Emerging trend of big data analytics in bioinformatics: a literature review. *Int J Bioinform Res Appl* 14(1–2):144–205
- Nassif AB, Capretz LF, Ho D (2012) Estimating software effort using an ANN model based on use case points. In: 2012 11th International Conference on machine learning and applications, 2: 42–47. IEEE.
- Nassif AB, Azzeh M, Capretz LF, Ho D (2013) A comparison between decision trees and decision tree forest models for software development effort estimation. In: 2013 Third International Conference on communications and information technology (ICCIT). IEEE, pp 220–224
- Nisar MW, Wang YJ, Elahi M (2008) Software development effort estimation using fuzzy logic A survey. In: Fifth International Conference on fuzzy systems and knowledge discovery, 1: 421–427
- Nitze A, Schmietendorf A, Dumke R (2014) An analogy-based effort estimation approach for mobile application development projects. In: 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on software process and product measurement. IEEE, pp 99–103
- Oliveira AL, Braga PL, Lima RM, Cornélio ML (2010) GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Inf Softw Technol* 52(11):1155–1166
- Papatheocharous E, Andreou AS (2009) Classification and prediction of software cost through fuzzy decision trees. *International Conference on enterprise information systems*. Springer, Berlin, Heidelberg, pp 234–247
- Patra HP, Rajnish K (2018) Development of software effort estimation using a non-fuzzy model. *Int J Comput Syst Eng* 4(1):17–22
- Peixoto CEL, Audy JLN, Prikladnicki R (2010) Effort estimation in global software development projects: preliminary results from a survey. In: Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on. IEEE, pp 123–127
- Ponalagusamy R, Senthilkumar S (2011) Investigation on time-multiplexing cellular neural network simulation by RKAHeM (4, 4) technique. *Int J Adv Intell Paradig* 3(1):43–66
- Pospieszny P, Czarnacka-Chrobot B, Kobylinski A (2018) An effective approach for software project effort and duration estimation with machine learning algorithms. *J Syst Softw* 137:184–196
- Qi F, Jing XY, Zhu X, Xie X, Xu B, Ying S (2017) Software effort estimation based on open source projects: case study of Github. *Inf Softw Technol* 92:145–157
- Rao PS, Kumar RK (2015) Software effort estimation through a generalized regression neural network. In: Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1, pp 19–30. Springer, Cham
- Reynolds J, Ahmad MW, Rezguy Y (2018) Holistic modelling techniques for the operational optimisation of multi-vector energy systems. *Energy Build* 169:397–416
- Saeed A, Butt WH, Kazmi F Arif M (2018) Survey of software development effort estimation techniques. In: Proceedings of the 2018 7th International Conference on software and computer applications, pp 82–86
- Sarro F, Petrozziello A (2018) Linear programming as a baseline for software effort estimation. *ACM Trans Softw Eng Methodol (TOSEM)* 27(3):1–28
- Satapathy SM, Rath SK (2014) Class point approach for software effort estimation using various support vector regression kernel methods. In: Proceedings of the 7th India Software Engineering Conference, pp 1–10
- Satapathy SM, Rath SK (2017) Empirical assessment of machine learning models for effort estimation of web-based applications. In: Proceedings of the 10th Innovations in Software Engineering Conference, pp 74–84
- Shepperd M, Schofield C (1997) Estimating software project effort using analogies. *IEEE Trans Softw Eng* 23(11):736–743
- Sheta AF, Ayesha A, Rine D (2010) Evaluating software cost estimation models using particle swarm optimisation and fuzzy logic for NASA projects: a comparative study. *Int J Bio-Inspir Comput* 2(6):365–373
- Singh BK, Misra A (2012) Software effort estimation by genetic algorithm tuned parameters of modified constructive cost model for nasa software projects. *Int J Comput Appl* 59(9):22–26
- Stefanowski J (2006) An empirical study of using rule induction and rough sets to software cost estimation. *Fundam Inf* 71(1):63–82 (**Special issue on theory and applications of soft computing (TASC04)**)
- Tosun A, Turhan B, Bener AB (2009) Feature weighting heuristics for analogy-based effort estimation models. *Expert Syst Appl* 36(7):10325–10333
- Trendowicz A, Jeffery R (2014) Classification of Effort Estimation Methods. In: Software project effort estimation. Springer, Cham, pp 155–208
- Walkerden F, Jeffery R (1999) An empirical study of analogy-based software effort estimation. *Empir Softw Eng* 4(2):135–158
- Wen J, Li S, Tang L (2009) Improve analogy-based software effort estimation using principal components analysis and correlation weighting. In: 2009 16th Asia-Pacific Software Engineering Conference, pp 179–186. IEEE
- Wen J, Li S, Lin Z, Hu Y, Huang C (2012) Systematic literature review of machine learning based software development effort estimation models. *Inf Softw Technol* 54(1):41–59
- Whigham PA, Owen CA, Macdonell SG (2015) A baseline model for software effort estimation. *ACM Trans Softw Eng Methodol (TOSEM)* 24(3):1–11
- Zare F, Zare HK, Fallahnezhad MS (2016) Software effort estimation based on the optimal Bayesian belief network. *Appl Soft Comput* 49:968–980
- Zarei A, Asl BM (2020) Automatic classification of apnea and normal subjects using new features extracted from HRV and ECG-derived respiration signals. *Biomed Signal Process Control* 59:101927
- Zhang D, Tsai JJP (2003) Machine learning and software engineering. *Softw Qual J* 11(2):87–119
- Zhang W, Yang Y, Wang Q (2015) Using Bayesian regression and EM algorithm with missing handling for software effort prediction. *Inf Softw Technol* 58:58–70
- Zhang H, Ren JX, Kang YL, Bo P, Liang JY, Ding L, Kong WB, Zhang J (2017) Development of novel in silico model for developmental toxicity assessment by using naive Bayes classifier method. *Reprod Toxicol* 71:8–15

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.