International Journal of Computer Applications (0975 – 8887)
Volume 184– No.21, July 2022

A Systematic Literature Review on Effort Estimation in Agile Software Development using Machine Learning Techniques

Pranay Tandon
Research Scholar,
School of Computer Science & IT, Devi Ahilya University,
Indore, M.P., India

Ugrasen Suman, PhD
Professor,
School of Computer Science & IT, Devi Ahilya University,
Indore, M.P., India

Maya Rathore, PhD
Associate Professor,
Christian Eminent College,
Indore, M.P., India

ABSTRACT
Agile software development is a way of frequent or continuous delivery of software. Nowadays many software industries have adopted agile for software development. The predictability and stability of traditional methods were replaced with flexibility, adaptability and agility to generate maximum value with collaboration and interaction, as quickly as possible. Effort estimation is the focused area in agile software development to achieve customer collaboration, respond to change and deliver a working software on time. Machine learning is an advanced tool to obtain effort estimation with available project data and widely used in IT industries to get accurate estimations. In this paper, the findings are reported through systematic literature review that aimed at identifying the applicability, limitations and individual result of most used machine learning techniques for effort estimation in agile software development with the help of 3 research questions. Also, suggested attributes of a robust machine learning model are discussed to achieve more accurate effort estimation. Conclusion of paper can help researchers and IT consultants in building a ML model considering the applicability, results and limitations of ML techniques.

Keywords
Agile software development, effort estimation, machine learning, systematic literature review, techniques, methods, limitations, model, deep learning.

1. INTRODUCTION
Agile software development (ASD) is a set of frameworks and practices based on the values and principles expressed in the agile manifesto and 12 principles behind it. Many frameworks are used in ASD such as, Scrum, Extreme Programming, Kanban, Crystal, Feature Driven Development etc. There are various practices such as, pair programming, test driven development, daily stand-up meetings, planning sessions, continuous integration, sprints etc. [32]. There are many research dimensions in ASD such as, estimation and planning, development, testing, enhancement and maintenance and project management. Estimation and planning have become an important dimension for practitioners and academicians.

Estimation is one of the most challenging areas of project management. For decades, project professionals have struggled with correct estimation of effort, cost and duration of initiatives that is required for development of schedules and budgets. The difficulty lies in forecasting those parameters at the initial stages of the project life cycle when boundaries of every initiative need to be established and when uncertainty regarding functionality of the final product is substantial. Oftentimes, limited knowledge about influencing factors and risks which may occur, pressure from client or management, and legacy software estimation techniques based on expert judgment may lead to imprecise and usually overoptimistic estimates. As a result, they may severely impact delivering project outcomes within a defined time frame, budget and of acceptable quality [1]. A good estimation leads to efficient planning, improved resource management, on time delivery, stronger client relationship, standard quality of product and better reputation of organization,

Many effort estimation techniques in ASD have been used such as, planning poker (PP), expert judgement, function points (FP), analogy, dis-aggregation, and algorithmic approach. Machine learning (ML) is also one of the important approaches nowadays for estimation and tool design [2]. ML is an application of Artificial Intelligence (AI) that provides systems the ability to learn and improve from experience without any explicit programming. ML focuses on the development of such computer programs that access data and use it to learn for themselves. The process of learning begins with observations or data such as, examples, direct experience or instructions, to look for patterns in data and make better decisions in future, based on the examples provided.

ML algorithms are useful in estimation because of its power of reasoning and learning process. ML algorithms can be used for extracting useful knowledge from data through the process of automated learning based on input. ML algorithms can mimic the human learning process up to a certain point and suitable for modelling complex problems such as, effort estimation, which can hardly be programmed [1].

There exist various research issues associated such as : small data set, outliers, categorical features, missing values etc. and these contexts impact characteristics of ML techniques. ML techniques require optimized and large enough data set, model building, model, training and testing, correct feature selection and inclusion of all interrelated factors that can influence the process. ML techniques have different strengths and weaknesses and work differently on various data sets. Combining two or more ML techniques may have the potential ability to enhance the power of estimation model [2].

This paper is organized as follows. Section 2 presents a brief background and summarizes the related work. Section 3 describes the research method, which is used in this study. Section 4 presents the results from systematic literature review (SLR), Section 5 presents the discussion and answers of research questions. Finally, section 5 provides concluding remarks.

2. LITERATURE SURVEY

ML is a new dimension of effort estimation. In recent years, ML based method has been receiving increasing attention in software development effort estimation research. Many ML techniques have been reviewed in literature survey. Different types of neural networks are compared such as, GRNN, Probabilistic neural network (PNN), GMDHPNN and cascade-correlation neural network (CCNN). The learning process in CCNN is quick. CCNN is found best among them. limitations are assumptions of team's initial project velocity value, project type is not available in data and small data set. As future work other machine learning techniques such as SGB, RF etc. can be implemented along with Story point approach (SPA) [5].

Ensemble-based model is proposed. It is proved that ensemble-based prediction is better than other prediction techniques. As limitations this approach and model is limited to the dataset from a specific organization and some predictive algorithms in the ensemble provided better predictions than this ensemble algorithm. For improvement it can include human experts in ensembles and consider developing efficient optimization approaches at the project portfolio level [6].

DT is explored with planning poker and it is found that planning poker with decision tree and planning poker with logistic model tree are better than planning poker alone. Multiple ML algorithms or ensemble- based algorithm can be used with planning poker [7]. Natural language processing (NLP) is used for text classification and then autoencoders used for estimation. In future text classification can be improved with advanced NLP process and autoencoders can be optimized [8]

Different ML techniques such as decision tree, SGB and random forest used with SPA and compared, SGB is found best among them. Limitations of this research are assumptions of team's initial project velocity value and small size data set. Further extreme learning machine and BN on the SPA-related dataset can be used [9]. ANN, SVM, K-star and linear regression algorithms are evaluated and found SVM has the best prediction accuracy [10].

Term frequency – Inverse document frequency (TF-IDF) and doc2vec text vectorization is used with SVM and Gaussian Naive Bayes ML methods and concluded that better estimations can be obtained than COCOMO. Large data set can be used in future [11]. According to one systematic literature review (SLR) fuzzy logic and ANN are the most used ML methods, LOC is most used size metrics, NASA and ISBSG is most used data set, cross validation is most used validation method, but this SLR includes research papers from 2015 to 2017 only this range can be increased [12]. Doc2vec is used for text vectorization and ANN is used for estimation. This approach is better than other ML methods but unable to surpass human estimations. Hence More research needs to be done on the text processing algorithms to improve results [11].

BN model is also used for estimation with more accuracy than other ML approaches. proposed model is relatively small, simple and all the input data are easily elicited, so that the impact on agility is minimal. The model predicts task effort, and it is independent of agile methods used. still all influencing factors were not used in this approach [19]. Main limitation of the BN model is validation for future research, it intends to validate the model in two stages: node probability tables validation and model validation. It will define more scenarios and will compare, in collaboration with experts [14]. Estimation based on Support vector regression (SVR) optimized by grid search method. Grid search method improves greatly the performance of SVR-RBF (radial basis function). As future work different validation methods and different datasets can be used [15]. SPA optimized using ML techniques adaptive neuro-fuzzy modelling, GRNN and Radial basis function neural networks (RBFNN) and found performance is approximately similar with anfis, newgrnn, newrb and newrbe functions. Limitations of this research are automation of user stories to story points is not possible, small data set and assumptions of velocity of team [16].

SPA is used with ANN, SVR and RF. As conclusion RF technique is the best preferred technique compared to ANN and SVR. Small data set and unavailability of agile story points are limitations [17]. A decision support tool produced using SVM, ANN and generalized linear models. an ensemble-based averaging is used in this proposed model and found better than other ML approaches. Limitations are all influencing factors are not present in data set, SVR outperformed in many cases which can impact averaging. As future work cross industry data set can be used [18].

Evolutionary cost sensitive deep belief network (ECS-DBN) is developed. This ECS-DBN model is relatively small and simple, and all the input data are easily elicited. The application can be extended to other deep learning methodologies with higher dimensional data for better performance [20]. ANN-feedforward back-propagation neural network, CCNN and Elman neural network are compared. Effort estimation of feedforward back-propagation network is better than 2 others. In future more ANN can be compared using large enough data set [21].

NB, logistic regression (LR), and RF are used and compared. As conclusion RF achieved the best performance among these. This research can be extended with the use of data mining ML methods [22].

MLP, GRNN, RBFNN and CCNN are compared and found CCNN is the best. Limitations are that performance evaluation is not done with Mean Magnitude of Relative Error (MMRE) and cross company data is used but within company data is not in ISBSG data set. Future work will focus on conducting the comparison of models using within-company projects. Leave one out validation technique and other performance evaluation criteria can also be considered in the [23]. Estimation done using multi-layered feed forward neural network which is given training with back propagation training method. Proposed model is better than COCOMO. This model will be extended by integrating the proposed approach with genetic algorithm techniques [24].

SLR is done on ML methods Ordinary least squares regression, Ridge regression least absolute shrinkage and selection operator regression, elastic-net regression, least angle regression, classification and regression tree, analogy-based estimation (ABE), SVR, deep neural networks, ANN, bootstrap aggregating, adaptive boosting, RF, and gradient boosting machine. Ensemble learning algorithms based on the principle of bootstrap aggregating, for example, Bagging and RF, performed the best overall over the 13 datasets. ABE appeared to be the highest-performing non-ensemble learning algorithm [25].

Genetic algorithms such as SVR, MLP neural networks and decision tree are used for feature selection and ML parameter optimization. This method achieved the best performance in terms of PRED in all the data sets. the multiple additive regression trees models outperformed this model regarding the MMRE in the NASA dataset. In future another optimization method, namely, Particle swarm optimization (PSO), for simultaneous feature selection and parameter optimization can be used for software effort estimation [26].

ANN, CBR and regression models are compared using function point approach and concluded ANN is the best. All influencing factors can be used in future ([27]. Long short-term memory (LSTM) is used with regression as activation function and recurrent highway network. This Proposed approach outperformed the existing technique using TF-IDF in estimating the story points. Data set is small, this is the limitation of this research. Feature selection can be improved in future [28]. LSTM and recurrent network are used for estimation and compared against the Bag of words and doc2vec techniques, this approach has improved in mean absolute error. Scope of influencing factors can be improved [29].

SLR performed on ANN. As conclusion COCOMO dataset is the most utilized dataset, Feed forward neural network (including MLP and Back propagation) is the most used technique, MMRE is most used accuracy measure. Future study may focus on working and exploring more about higher-order neural networks and deep learning neural networks and their application in effort estimation [30].

3. RESEARCH METHOD

Systematic literature review (also referred to as a systematic review) is a form of secondary study that uses a well-defined methodology to identify, analyze and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable (Kitchenham et al., 2007).

3.1 Research Questions

In order to develop a complete understanding of applicability, result and limitations of ML techniques used in effort estimation in ASD the following research questions (RQ) are formulated:

RQ1 : What is the applicability of ML technique for software effort estimation?

RQ2 : What is the effort estimation result of ML technique?

RQ 3: What are the limitations or improvement area of ML technique?

3.2 Data Sources

There are several sources of academic databases. The following databases are chosen, which are considered as mainstream venues for ASD and ML:

- IEEE Xplore (www.ieeexplore.ieee.org)
- Elsevier Science Direct (www.sciencedirect.com)
- SpringerLink (www.springerlink.com)
- Other standard database

3.3 Data Retrieval

To search all the ML techniques used in effort estimation in ASD, search string is used as follows:

((“effort”) AND (“estimation” OR “prediction”) AND (“in agile software development using”) AND ("machine learning" OR "deep learning" OR "artificial intelligence" OR “neural network” OR “natural language processing” OR “ensemble learning” OR “soft computing”) AND (“techniques” OR “algorithms” OR “models” OR “approaches” OR “methods” OR “frameworks”))

3.4 Studies Selection

Primary studies were included according to the following criteria:

- Were written in English.
- Were available online.
- Were published in last 10 years (mostly).
- Have discussed ML techniques and its use in effort estimation.
- Have performed the estimation process in agile methodology.

Articles were excluded if they:

- Were duplicate or repeated studies.
- Were not directly related to the objective of the research.

The electronic databases mentioned above, are searched using the search string. Search string has been adapted according to the database. In the first review, the title and abstract of the paper is reviewed to decide whether to include it or not. papers selected in first review are again reviewed by reading their introduction, few pages, and conclusion. Then the subset of the papers, which were found relevant, was selected. In the final review, whole papers were read and verified whether they are satisfying the inclusion criteria. The result of applying search string, first review, and second review, and final selection is shown in Table 1.

Table 1: Resources and selections of studies.

Resources	SearchResults	Primary Selection	Secondary Selection	Final Selection
IEEE Xplore	618	114	47	10
Elsevier Science Direct	595	108	35	9
Springer Link	2210	480	54	5
Others	297	87	18	6
Total	3720	789	154	30

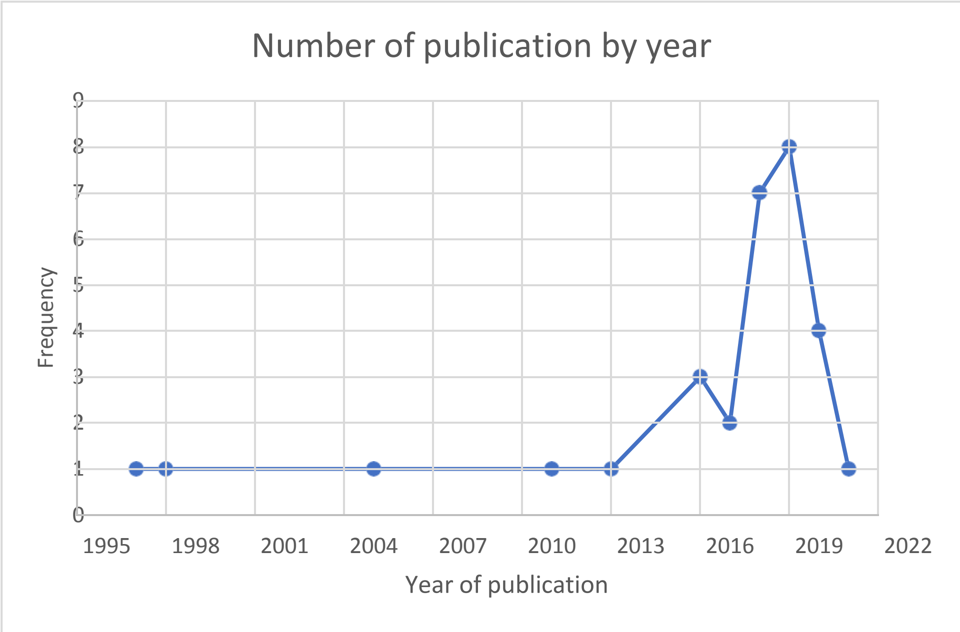

Figure 1: Frequency of publications by year

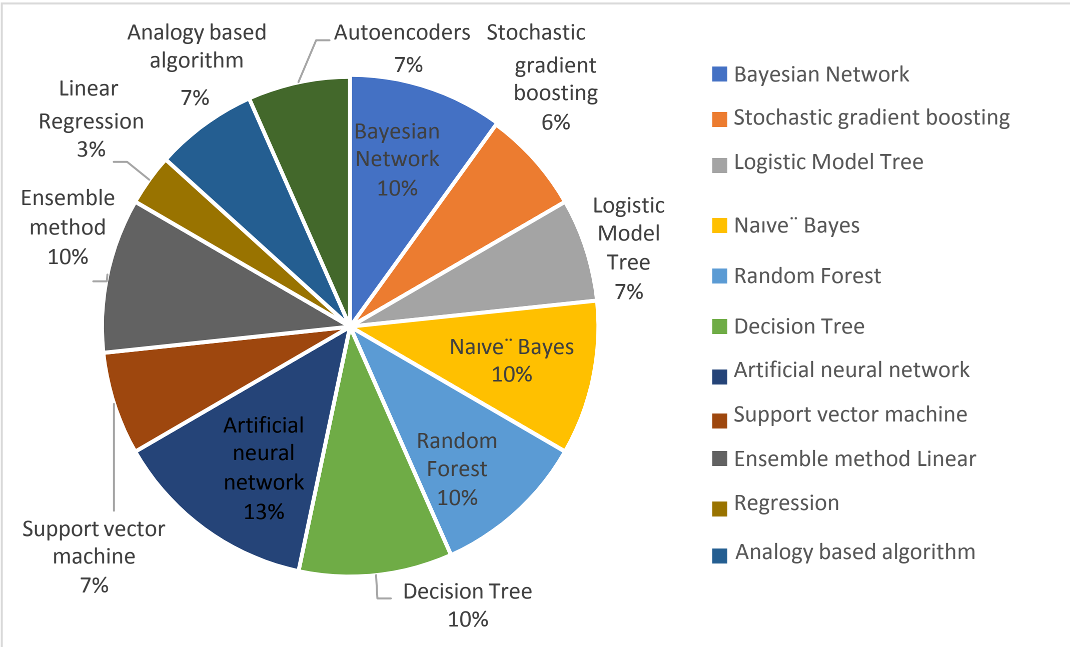

Figure 2: Distribution of primary studies according to ML techniques

3.5 Data Extraction

30 papers were selected after final selection process. Microsoft Excel was used for data extraction. Data extraction form contains the following information: title, authors, publication year, database, ML techniques used, result and limitations of ML technique. For 30 selected papers, data was recorded. Then, qualitative analysis was performed to categorize the ML techniques applicability, result and limitations, this was reported in the result section.

3.6 Threats to Validity

Limited databases are searched for constructing SLR due to budget and time constraints, but most of the relevant sources of ML and Agile research are covered. Most of the keywords used to represent ML are included. The search string has been slightly modified according to the search engines to reduce the number of irrelevant studies. Search process for extracting relevant studies are applied sincerely, thoroughly and systematically. It might be possible that few papers may have not been considered due to rising number of studies in this research field. For ensuring the reliability of search process, the first author has applied the search process and the second author has cross checked the results of search from time to time.

4. RESULTS

4.1 Year wise distribution of studies

Figure 1 shows year wise distribution of publications. It can be seen that the number of papers from 1996 to 2012 were less than papers from 2013 to 2020. 25 studies (i.e., 83%) were published after 2013. This shows that the interest in ML techniques for effort estimation in ASD has increased after 2013.

4.2 ML technique wise distribution of studies

Figure 2 depicts the distribution of primary studies according to ML techniques. The percentage of ANN (i.e., 13%) and ensemble method, BN, RF, DT, NB (i.e., 10%) reveals that interest of researchers as well as practitioners has been constantly increasing in using ML techniques to find effort estimation in ASD.

5. DISCUSSION

With this literature review, some limitations of research in effort estimation in ASD using ML can be found. Some of the data in data sets are missing such as initial velocity, project type, number of sites. There is an assumption value taken for missing data. In many research Data sets are small and this leads to lack of generalization, data imbalance and difficulty in optimization. Due to small data sets and missing values, other issues in ML models can be faced such as outliers, improper split of train and test data, over fitting, measurement errors, sampling bias etc.

In previous research many ML methods have been evaluated and compared, but few research works showed evaluation of almost all relevant ML methods and their comparison with proper validation. In some research result of ML models is compared to basic COCOMO model's result. ML methods performs differently with different data sets and there is not a single ML algorithm which outperforms with all data sets.

In effort estimation in ASD, there are many influencing factors. In previous research all influencing factors are not used and the impact of these factors on estimation is not considered. There can be many influencing factors in any ASD project such as, project domain, performance, configuration, data transaction, complex processing, operation ease, multiple sites, security, type of project, quality requirements, hardware and software requirements, communication skill, team skill, managerial skill, working time, experience of project team, technical ability etc.

As future work, large and cross industry data sets can be used in ML model for effort estimation in ASD. Missing values should be obtained through some standard process and optimize dataset accordingly. More ML techniques can be explored, two or more ML techniques can be used for model creation to use ensemble-based ML model. Expert judgement can be combined with one or more ML model to achieve advantages of expert's experience and ML technique's prediction.

Deep learning and higher order neural network can also be explored for prediction and autoencoders can be used to optimize and best fit data. NLP also applied to extract keywords from statement of user story and estimate story points according to keywords, in this area more research needs to be done on the text processing algorithms.

Some research gaps are founded with this literature review in field of effort estimation in ASD using ML techniques are small data sets, missing values, missing attributes which can impact the estimation, comparison of proposed evaluation with basic estimation process and very few research works are done on deep learning and NLP.

In this paper the applicability, individual results and limitations of most used ML techniques are found and presented for estimation in ASD, in compare to previous related works, more details are added in tabular form as answers of 3 main research questions. Results can positively impact selection of ML techniques in building a ML model and it can improve accuracy of effort estimation in ASD.

Review has been done on systematically selected papers following standard criteria for selection, and all the results presented in the form of tables and figures for better understandability. As per the current scope of research questions the analysis is sufficient and future work is also defined in the paper.

What is the applicability of ML technique for software effort estimation?

Table 2: ML techniques applicability for effort estimation

Sr.No.	ML Technique	Applicability for effort estimations
1	Bayesian Network	A BN is a directed acyclic graph (DAG) that represents "a joint probability distribution over a set of random variables"
2	Stochastic gradient boosting	Boosting implies applying the function in iterative fashion in a series and consolidate the yield of each function with a weighting coefficient keeping the goal in mind the end goal to minimize the aggregate prediction error and improve the accuracy.
3	Logistic Model Tree	LMT algorithm is for supervised learning tasks, which combines linear regression, LR and tree induction. LMT produces classification models that are more accurate than those produced by Classification and Regression Trees (CART) and simple LR.
4	Naıve¨ Bayes	NB classifier is a probabilistic classifier based on Bayes' theorem with an assumption of independent features given the class
5	Random Forest	RF consists of multiple decision trees and outputs the class that is the statistical mode of the classes output by individual trees. Random Forest is suited for learning from large datasets and avoids over fitting better than decision trees.
6	Decision Tree	DT illustrates the prediction of a dependent variable using a set of predictor variables. Decision trees are fitting well to the training data when the tanning set is not linearly separable.

7	Artificial neural network	Multilayer perception, it is one of the most used supervised model, it is a multi-layers network of neurons, connected by a feed-forward mode. a NN can be seen as a complex computation function that passes data through the network to the output layer to expand the solution.
8	Support vector machine	SVM depends on the declaration of a decision plane, which defines decision margins. SVM acts as a supervised classifier model that implements the classification task by building hyper-plane in a multidimensional space.
9	Ensemble method	Ensemble methods use the idea of combining several predictive models to get higher quality predictions than each of the models could provide on its own.
10	Linear Regression	This Algorithm is used to express the data and find the correlation between the dependent variable and one or more independent nominal, comma, or level variables.
11	Analogy based algorithm	Analogy-based estimation (ABE) applies the k-nearest neighbor algorithm to estimating the target variable. the estimated effort value for a new software project can be defined as the total amount of effort used in past similar software projects. [25].
12	Autoencoders	Autoencoder is nonlinear adaptive feature extraction technique that consists of a NN trained with unsupervised algorithm. consists of 3 components: encoder, code, decoder. encoder compresses the input, produces the code, decoder then reconstructs the input only using this code.

What is the effort estimation result of ML technique?

Table 3: ML technique's effort estimation result

Sr.No.	ML Technique	Data set	Performance measure	Result
1	Bayesian Network	Agile project dataset	MMRE, Pred (m), MAE,RMSE	MMRE 6.21 for 160 tasks. The MMRE values show that there are no occasional large estimation errors.
2	Stochastic gradient boosting	Agile Project dataset	MAE, MMER, Pred (x)	MMER 0.1632, Pred (25) 85.7143. Using SGB based model the accuracy is observed to be quite high.
3	Logistic Model Tree	Agile Project dataset	MMRE	MMRE 125.26 %. LMT alone is not enough for accurate effort estimation but LMT with PP has a good accuracy.
4	Naıve¨ Bayes	NASA, Agile Project dataset	MMRE, AUC, CA, Precision and Recall	MMRE 0.592, AUC 96%, CA 83%, precision 88%, recall 93%. Naïve Bayes produces better result when using doc2vec for text vectorization.
5	Random Forest	Agile Project dataset, NASA	MMRE, MSE, AUC, CA, Precision and Recall	MMRE 0, MSE 0.26, AUC 96%, CA 86%, precision 93%, recall 83%. RF is fast to build and to predict. RF technique can be of appropriate use for the analysis of complex datasets.
6	Decision Tree	Agile Project dataset	MMRE, MAE, MMER, Pred (x)	MMRE 92.32%, MMER 0.3820, pred(25), 38.0952. decision tree shown very good accuracy of estimation and with planning poker it shown better accuracy.
7	Artificial neural network	Usp05-tf, Agile Project dataset,	E, MMRE, MSE	MAE 2.7826, MMRE 1.0932, MSE 12.41. The ANN models appears capable of effectively capturing the parameters influencing productivity but does so at the expense of comprehensibility
8	Support vector machine	Usp05-tf, Agile Project dataset, ISBSG	MAE, MMRE	MAE 2.5958, MMRE 0.13.SVM has outstanding ability to handle complex dependencies within the heterogeneous data.
9	Ensemble method	Agile Project dataset, ISBSG	MAE, MMRE	MAE 8.167, MMRE 91.75 %. combining multiple solo algorithms into a stacked ensemble of multi-algorithms by maximizing both the potential performance of each individual algorithm and to maximizing the level of diversity of the selected individual algorithms has great potential to offer a more accurate estimation for software effort as compared with other different stacking approaches.

10	Linear Regression	Usp05-tf	MAE	MAE 5.9965. Linear regression has the highest MAE value comparing to ANN, SVM and k Star. this is caused by fact that the distribution of the effort values is hard to be expressed as a linear equation.
11	Analogy based algorithm	Agile Project dataset, Albrecht	MMRE	MMRE 62%. ABE appeared to be the highest-performing non-ensemble learning algorithm in the comparisons.
12	Autoencoders	Agile Project dataset	cision, recall, f-measure.	Precision 0.81±0.11 Recall 0.89±0.06 F- measure 0.81±0.07 (using TF-IDF for text classification). Autoencoders can learn relevant features in various degrees of abstraction. Such methods can be used in text classification for finding useful semantics that may be used in the training of a classifier.

What are the limitations or improvement area of ML technique?

Table 4: ML technique's limitations

Sr. No.	ML Technique	Limitations (with respect to SEE)
1	Bayesian Network	BN tend to perform poorly on high dimensional data. BN cannot be used to model the correlation relationships between random variables.
2	astic gradient boosting	SGB continuously improves to minimize all errors. This can overemphasize outliers and cause over fitting. Must use cross-validation to neutralize. Computationally expensive - SGB often requires many trees (>1000) which can be time and memory exhaustive.
3	Logistic Model Tree	LR requires a large dataset and sufficient training examples for all the categories it needs to identify. LR can only be used to predict discrete functions. Therefore, the dependent variable of LR is restricted to the discrete number set. This restriction itself is problematic, as it is prohibitive to the prediction of continuous data.
4	Naıve¨ Bayes	If test data set has a categorical variable of a category that is not present in the training data set, the NB model will assign it zero probability and won't be able to make any predictions in this regard. It assumes that all the features are independent.
5	Random Forest	The main limitation of RF is that many trees can make the algorithm too slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained
6	Decision Tree	A small change in the data can lead to a large change in the structure of the optimal decision tree. For data including categorical variables with different numbers of levels, information gain in decision trees is biased in favor of those attributes with more levels. Calculations can get very complex, particularly if many values are uncertain and/or if many outcomes are linked.
7	Artificial neural network	ANN require processors with parallel processing power which is computationally expensive. When ANN gives a probing solution, it does not give a reason or justification for this solution. ANN usually require much more data than traditional machine learning algorithms
8	Support vector machine	Long training time for large datasets. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform. Choosing an appropriate Kernel function is difficult
9	Ensemble method	Ensemble methods are usually computationally expensive. They add learning time and memory constrains to the problem. The model that is closest to the true data generating process will always be best and will beat most ensemble methods. So, if the data come from a linear process, linear models will be much superior to ensemble models
10	Linear Regression	Assumption of linearity between the dependent variable and the independent variables. In the real world, the data is rarely linearly separable. If the number of observations is lesser than the number of features, linear regression should not be used Linear regression is very sensitive to outliers (anomalies). So, outliers should be analyzed and removed before applying linear regression to the dataset.
11	alogy based algorithm	Accuracy depends on the quality of the data with large data, the prediction stage might be slow, Sensitive to the scale of the data and irrelevant features. Require high memory. need to store all the training data, it can be computationally expensive. Requires analogues project for comparison which is rarely achievable in software development.

12	Autoencoders	Autoencoders are an unsupervised technique that learns from its own data rather than labels created by humans. This often means that autoencoders need a considerable amount of clean data to generate useful results, autoencoders are lossy, which limits their use in applications when compression degradation affects system performance in a significant way.

6. CONCLUSION

There were some limitations in earlier research in effort estimation using ML techniques in ASD such as small data sets, missing data, comparison with result of irrelevant ML techniques and consideration of less influencing factors in ML model. Also, individual limitations of ML technique can decrease the efficiency of ML model.

According to results, limitations and future works found in literature review an ensemble- based deep learning model can be developed with more relevant project influencing factors as features to overcome above mentioned research gap. a large enough data set can be used and missing values can be obtained with data optimization techniques. NLP also can be used to extract keywords from user story and assign user story point to each story according to keyword weight and calculation logic. After validation of this model finally this model can be compared with all deep learning models selected by performance criteria.

Result of this research can help researchers in field of ML usage in effort estimation in ASD, academicians for further studies and IT consultants and developer for building a ML model for estimation.

As future work more ML techniques used in recent research works will be included with large datasets and datasets of different type of agile projects to understand the applicability, limitations and result of ML techniques with respect of type of agile projects.

7. REFERENCES

[1] Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2017). An effective approach for software project effort and duration estimation with machine learning algorithms. Journal of Systems and Software.

[2] Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. Information and Software Technology, Vol. 54(1), pp. 41–59.

[3] Moløkken-Østvold, K. (2004). A survey on software estimation in the Norwegian industry. Proceeding International Software Metrics Symposium, pp. 208-219.

[4] Shepperd, M. & Schofield, C. (1996). Estimating software project effort using analogies. IEEE Transactions on Software Engineering, Vol. 23(11), pp. 736-743.

[5] Panda, A., Satapathy, S.M. & Rath, S.K. (2015). Neural Network Models for Agile Software Effort Estimation based on Story Points. International Journal of Artificial Intelligence and Neural Networks IJAINN, Vol. 5, Issue 2.

[6] Malgonde, O. & Chari, K. (2018). An ensemble-based model for predicting agile software development effort. Springer Science+Business Media, LLC, part of Springer Nature.

[7] Moharreri, K., Sapre, A.V., Ramanathan, J. & Ramnath, R. (2016). Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments. IEEE 40th Annual Computer Software & Applications Conference.

[8] Soares, R.G.F. (2018). Effort estimation via text classification and autoencoders. International Joint Conference on Neural Networks (IJCNN).

[9] Satapathy, S.M., & Rath, S.K. (2017). Empirical assessment of machine learning models for agile software development effort estimation using story points. Springer-Verlag London Ltd.

[10] Hammad, M., & Alqaddoumi, A. (2018). Features-Level Software Effort Estimation Using Machine Learning Algorithms. International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT).

[11] Ionescu, V.S., Demian , H., & Czibula, I.G. (2017). Natural Language Processing and Machine Learning Methods for Software Development Effort Estimation. Studies in Informatics and Control, 26(2) 219-228.

[12] Sharma, P., & Singh, J. (2017). Systematic Literature Review on Software Effort Estimation Using Machine Learning Approaches. International Conference on Next Generation Computing & Information Systems (ICNGCIS).

[13] Ionescu, V.S. (2017). An approach to software development effort estimation using machine learning. IEEE.

[14] Radu, L.D. (2019). Effort Prediction in Agile Software Development with Bayesian Networks. 14th International Conference on Software Technologies (ICSOFT 2019), pages 238-245.

[15] Zakrani, A., Nazm , A. & Marzak, A. (2018). Support Vector Regression Based on Grid Search Method for Agile Software Effort Prediction. IEEE.

[16] Rao, C.P., Kumar, P.S., Sree, S. R., & Devi, J. (2018). An Agile Effort Estimation Based on Story Points Using Machine Learning Techniques. Second International Conference on Computational Intelligence and Informatics, Advances in Intelligent Systems and Computing 712.

[17] Sree, S. R., Rao, C.P., & Mounika, M. (2017). An Early-Stage Software Effort Estimation in Agile Methodology Based On User Stories Using Machine Learning Techniques. International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887 Vol. 5 Issue XII.

[18] Pospieszny, P., Czarnacka-Chrobot, B. & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. The Journal of Systems and Software 137 (2018) 184–196.

[19] Srdjana, D., Stipe , C., & Mili, T. (2017). Bayesian

Network Model for Task Effort Estimation in Agile Software Development. The Journal of Systems & Software.

[20] Premalatha, H.M., & Srikrishna , C.V. (2019). Effort Estimation in Agile Software Development using Evolutionary Cost- Sensitive Deep Belief Network. International Journal of Intelligent Engineering and Systems, Vol.12, No.2.

[21] Bilgaiyan, S., Mishra, S. & Das, M. (2018). Effort estimation in agile software development using experimental validation of neural network models. Springer.

[22] Banimustafa, A. (2018). Predicting Software Effort Estimation Using Machine Learning Techniques. 8th International Conference on Computer Science and Information Technology (CSIT).

[23] Nassif, A.B., Azzeh, M., Capretz , L.F., & Ho, D. (2015). Neural Network Models for Software Development Effort Estimation: A Comparative Study. The Natural Computing Applications Forum.

[24] Rijwani, P., & Jain, S. (2016). Enhanced Software Effort Estimation using Multi Layered Feed Forward Artificial Neural Network Technique. Twelfth International Multi-Conference on Information Processing (IMCIP).

[25] Phannachitta, P., & Matsumoto, M. (2019). Model-Based Software Effort Estimation – A Robust Comparison of 14 Algorithms Widely Used in The Data Science Community. International Journal of Innovative Computing, Information and Control Vol. 15, Number 2.

[26] Oliveira, A.L.I., Braga, P.L., Lima, R.M.F., & Cornelio, M.L. (2010). Genetic Algorithm-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. Information and Software Technology 52 (2010) 1155–1166.

[27] Finnie, G. R., Wittig, G. E., & Desharnais, J.M. (1997). A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models. Elsevier.

[28] Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T., Ghose, A., & Menzies, T. (2015). A deep learning model for estimating story points", Journal of latex class files, Vol. 14, NO. 8, August 2015.

[29] Alsubhi, K.A. (2019). Effort Estimation in Agile Software Development Using Deep Learning Model. International Journal of Advances in Computer Science and Technology, Volume 8..

[30] Kumar, P.S., Behera, H.S., Kumari K., A., Nayak, J., & Naik, B. (2020). Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades. Computer Science Review 38 (2020) 100288.

[31] Kichenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Vol. 2.3 EBSE Technical Report, EBSE-2007-01. Software Engineering Group, School of Computer Science and Mathematics, Keele University, Keele, UK.

[32] Bilgaiyan, s., Mishra, s., & Das, M. (2016). A Review of Software Cost Estimation in Agile Software Development Using Soft Computing Techniques. IEEE 2016 2nd International Conference on Computational Intelligence and Networks (CINE) - Bhubaneswar, India.