Title page:

Machine learning based software fault prediction in software efficiency using Artificial neural network comparing with Logistic Regression for improved accuracy

YalakaNikhilReddy[1],E.K.Subramanian[2]

YalakaNikhilReddy[1]
Research Scholar,
BE Computer Science,
SIMATS Engineering,
SIMATS Institute of Medical and Technical Sciences,
SIMATS University, Chennai, Tamil Nadu, India, Pin code: 602 105.
nikhilreddyy1188.sse@saveetha.com

E.K.Subramanian[2]
Research Scholar,corresponding author,
Department of Computer Science Engineering,
SIMATS Engineering,
SIMATS Institute of Medical and Technical Sciences,
SIMATS University, Chennai, Tamil Nadu, India, Pin code: 602 105.
subramanianek.sse@saveetha.com

Keywords:Software Faults,Prediction,Software Efficiency,Machine Learning,Artificial Neural Network,.

ABSTRACT:

Aim: By applying machine learning approaches, namely Artificial Neural Networks (ANN) and Logistic Regression, the current study improves software defect prediction in software efficiency. Greater accuracy in software fault detection is the aim of the Artificial Neural Network model. **Materials and Methods:** The study's main source of data was the Kaggle dataset. In this experiment, two groups were used: Group 1 and Group 2, each of which had ten samples. While Group 1 utilized artificial neural networks, Group 2 utilized logistic regression. The total number of participants in this investigation was forty. Python was used for the sample size calculation as well as the subsequent statistical analysis and performance comparison. Alpha (a) was set at 0.05, beta (13) at 0.2, and statistical power (G-power) at 80% for the statistical analysis, which was carried out using clincalc.com. Accuracy value was the primary assessment criterion used in the inquiry to assess the algorithm's and the artificial neural network's effectiveness.Given that the algorithm iteration is ten and the sample size is 10, the significant result in this instance is $p=0.067$ ($p>0.05$), indicating statistical insignificance between the two groups.**Result:** Consequently, the prediction of software defects in software efficiency has been implemented through the use of artificial neural networks. Support vector machines attain an accuracy percentage of 61.90%, whereas artificial neural networks demonstrate an 84.1% rate of success. It seems that artificial neural networks are more accurate than logistic regression.**Conclusion:** Research indicates that artificial neural networks (ANN) outperform logistic regression (LR) in terms of accuracy. The study shows that, in terms of accuracy in software defect prediction, the ANN model performs better than LR through a thorough comparative examination. This highlights the need of choosing appropriate machine learning models and highlights the improved predictive power of ANNs. The results of the study indicate that software defect prediction can be made more accurately by employing artificial neural networks (ANNs) in place of previous techniques. This would significantly advance the field of software engineering.

Keywords:Software Faults,Prediction,Software Efficiency,Machine Learning,Artificial Neural Network,Logistic Regression.

INTRODUCTION:-

By utilizing machine learning techniques like Logistic Regression (LR) and Artificial Neural Networks (ANN), software fault prediction seeks to improve software efficiency and dependability by anticipating possible errors in software systems before they arise. Software faults, according to (“A Comparison of Some Soft Computing Methods for Software Fault Prediction” 2015) are flaws or problems in software code that may cause malfunctions or system failures. Anticipating these errors is essential for guaranteeing the dependability and efficiency of

software programs. Software fault prediction is extremely important in today's world, since software systems are commonplace and essential to many industries, including communication, transportation, healthcare, and finance. According to (“Iterated Feature Selection Algorithms with Layered Recurrent Neural Network for Software Fault Prediction” 2019), software errors can pose a risk to public safety, cause large financial losses, and harm one's reputation. Thus, keeping software systems' functionality and integrity intact requires the proactive detection and mitigation of software flaws. (“Towards an Ensemble Based System for Predicting the Number of Software Faults” 2017) Research on software defect prediction has a wide range of practical applications. Early fault detection, for example, can assist developers in prioritizing work related to code restructuring and debugging, resulting in software products that are more resilient and maintainable (Li et al., n.d.). Furthermore, predictive maintenance based on software fault prediction can reduce unplanned outages and maximize resource utilization in sectors like manufacturing and aviation where downtime can have serious effects (“A Systematic Review of Machine Learning Techniques for Software Fault Prediction” 2015).

Over the past five years, there has been a noticeable increase in the number of articles published on software defect prediction using machine learning approaches, according to databases from Google Scholar and IEEE Xplore. In particular, the IEEE Xplore database contains 1,800 publications from this time span, compared to 27,500 total articles displayed by Google Scholar.Four notable studies stand out among the field's most cited articles:In a thorough review of previous studies using machine learning to anticipate software faults, (“Website,” n.d.) highlighted a range of methodologies, datasets, and assessment metrics.In order to anticipate software defects, (Azam, Nouman, and Gill 2022) examined the performance of several machine learning methods, such as ANN and LR, and provided insights into the relative merits of each strategy.In comparison to conventional machine learning models, (Khalid et al. 2023) achieved higher performance with their innovative ensemble learning method for software defect prediction. (Rathore and Kumar 2020) demonstrated the potential influence on operational efficiency and practical significance of software failure prediction by focusing on its application in industrial settings.The paper by (Azam, Nouman, and Gill 2022)in my opinion, is the most extensive and instructive because it offers a comprehensive review of the research environment, including methodology, difficulties, and potential paths for future research in software fault prediction.

The lack of comparative studies comparing Logistic Regression (LR) and Artificial Neural Networks (ANN) for software defect prediction is the focus of this work. Our team seeks to ascertain the optimal strategy for augmenting software efficiency and dependability by utilizing its proficiency in machine learning, software engineering, and data analysis. By means of thorough experimentation, we will compare the performance of ANN and LR models using relevant datasets and evaluation measures. The results will benefit software engineering practitioners by providing useful insights in addition to advancing software failure prediction

tools. This study emphasizes how important it is to choose appropriate machine learning techniques in order to reduce software errors and enhance system performance as a whole
.

MATERIALS AND METHODS :-

Completed the given tasks with the aid of the Saveetha School of Engineering Open Source Lab. Out of all the groups that have been found, there are two. Groups 2 and 1 used artificial neural networks and logistic regression, respectively. Using a dataset with a sample size of twenty items that included a variety of faulty items, the Artificial Neural Network approach and the Logistic Regression method were applied at different intervals. In the computation, 0.05 alpha, 0.2 beta, and a 95% confidence interval that represented 80% of the G-power value were employed.

The inefficiencies of the software were found in the real-time dataset. The predicted defect dataset served as the input for the proposed investigation. the "software defect prediction, 2018" spreadsheet file was taken from kaggle.com." To improve accuracy (%), the attributes "lines operand," "lines of code," and "faults" were mostly used.The description of the fault prediction dataset is primarily relevant. Feature extraction and cleaning will come after the dataset's initial pre-processing. Evaluation of artificial neural network performance was made possible by the obtained data set. Calculate how flawed it is. To train the database for fault prediction, use the attributes for prediction through the use of an artificial neural network. Two-thirds of the dataset was used for training and twenty percent was used for testing during preprocessing.

ARTIFICIAL NEURAL NETWORK ALGORITHM:-

A computational model called an artificial neural network (ANN) is modeled after the architecture and operation of biological neural networks seen in the human brain. Layer-by-layer network architectures (ANNs) consist of interconnected nodes that apply activation functions to incoming data in order to process information. Artificial neural networks (ANNs) use techniques like backpropagation to minimize prediction errors during training by adjusting the weights of connections between nodes. Natural language processing, image identification, predictive modeling, and software defect prediction are just a few of the applications that benefit greatly from the superior ability of artificial neural networks (ANNs) to capture intricate, nonlinear patterns in data. If appropriate regularization techniques are not used, ANNs may become overfitting and need a large amount of computational resources for training. ANNs are quite capable of addressing many different types of machine learning and artificial intelligence problems, even in spite of these difficulties.

LOGISTIC REGRESSION ALGORITHM:-

A statistical technique for binary classification problems, such as software failure prediction, is called logistic regression (LR). It simulates the connection between one or more predictor factors and a binary outcome variable. By applying the logistic function to the linear combination of predictors, LR calculates the probability of the outcome variable (such as the existence or absence of a fault) depending on the input features. After producing probability between 0 and 1, cases are categorized into one of the two groups. Because of its interpretability, computational efficiency, and simplicity, LR is preferred. Nevertheless, it makes the assumption that predictors and the log-odds of the result have a linear relationship, which may not always capture intricate nonlinear patterns in the data.

STATISTICAL ANALYSIS:-

Statistical analysis is done with IBM SPSS software. The importance of the differences in prediction accuracy will be assessed by the application of techniques like cross-validation and hypothesis testing. Further, to determine the elements impacting each algorithm's sales forecasts, a feature importance study will be carried out. We seek the optimal approach for improving the accuracy of software failure prediction models by a comprehensive statistical analysis.

RESULTS:-

The results of the study indicate that there was a significant difference in the prediction accuracy of software fault between the Artificial Neural Network (ANN) and Logistic Regression models. Specifically, the ANN model's accuracy rate was 84.1%, whereas the Logistic Regression model's was 61.9%. According to the Independent Sample t-test, there was a statistically significant difference in accuracy ($p < 0.01$). This suggests that the ANN technique may perform better in terms of accuracy-related software failure prediction.

DISCUSSION:-

In this work, we examined the usefulness of machine learning methods for software defect prediction in improving software efficiency, particularly Artificial Neural Network (ANN) and Logistic Regression (LR). Based on a number of performance criteria, including precision, recall, and F1-score, we discover that ANN performs significantly better than LR in terms of accuracy ($p < 0.05$).

Our analysis of the relative effectiveness of ANN and LR is consistent with earlier research conducted by (Mehmood et al., n.d.) (2023) and ("An Efficient Convergence-Boosted Salp Swarm Optimizer-Based Artificial Neural Network for the Development of Software Fault Prediction Models" 2023), which also found that ANN performed better in predictive modeling tasks. Nonetheless, it is imperative to recognize divergent results, since certain research, such as (Jiang, Cukic, and Ma 2008), has documented similar performance for ANN and LR. Despite these differences, there is a growing body of research supporting ANN over LR for predictive tasks because of its superior capacity to identify intricate patterns in the data(Babu, Sivasubramanian, and Natarajan 2022).

While our study offers insightful information, it is not without flaws. The dataset that was employed may not accurately reflect the variety of software systems and malfunction scenarios, which is one of its limitations("Software Defect Prediction Based on Enhanced Metaheuristic Feature Selection Optimization and a Hybrid Deep Neural Network" 2021). Furthermore, the model setups and hyperparameter selections may create bias and affect how broadly applicable our findings are. ("Software Fault Prediction: A Literature Review and Current Trends" 2011)Furthermore, it might be difficult to effectively describe failure behavior in software systems because of their intrinsic complexity, which could affect how well ANN and LR perform in comparison.

Subsequent investigations may pursue multiple approaches to mitigate the constraints and expand upon the outcomes of this investigation. First, exploring the use of other ensemble methods or machine learning approaches may improve forecast accuracy even more. To encourage the use of predictive models in actual software development environments, additional efforts should be made to make them easier to read. More reliable and broadly applicable models may also be produced by investigating cutting-edge methods for feature engineering and dataset creation specifically suited to software fault prediction.

CONCLUSION:-

To sum up, this study compared machine learning-based methods for software failure prediction that use logistic regression (LR) and artificial neural networks (ANN) to increase accuracy. The results showed that the ANN model outperformed the logistic regression model in terms of accuracy in predicting software faults. The ANN model was found to have an accuracy rate of 84.1%, which was higher than the logistic regression model's accuracy rate of 61.9%. These findings firmly imply that using ANN approaches to software engineering applications in the real world has great potential for greatly increasing software fault prediction accuracy.

DECLARATIONS:-

Conflict of interests

There is no conflict of interest in this work.

Authors Contribution

Writing the manuscript and handling data analysis and collecting fell to author YNR. Author EKS was responsible for the conceptualization, data validation, and critical assessment of the text.

Acknowledgements

The Saveetha School of Engineering, Saveetha Institute of Medical and Technical Science (formerly Saveetha University) is acknowledged by the authors for giving the support required to successfully finish this work.

Funding

We acknowledge the following organizations for their financial support, which allowed us to finish this study.

1. INautix Technologies, India.
2. SIMATS Engineering.
3. SIMATS University.
4. SIMATS Institute of Medical and Technical Sciences.

REFERENCES:-

"A Comparison of Some Soft Computing Methods for Software Fault Prediction." 2015. *Expert Systems with Applications* 42 (4): 1872–79.

"An Efficient Convergence-Boosted Salp Swarm Optimizer-Based Artificial Neural Network for the Development of Software Fault Prediction Models." 2023. *Computers & Electrical Engineering* 111 (October): 108923.

"A Systematic Review of Machine Learning Techniques for Software Fault Prediction." 2015. *Applied Soft Computing* 27 (February): 504–18.

Azam, Muhammad, Muhammad Nouman, and Ahsan Rehman Gill. 2022. "Comparative Analysis of Machine Learning Technique to Improve Software Defect Prediction." *KIET Journal of Computing and Information Sciences* 5 (2). https://doi.org/10.51153/kjcis.v5i2.96.

Babu, Rathish Babu Thirukonda Krishnamoorthy, Suresh Sivasubramanian, and Sankarram Natarajan. 2022. "MLPNN-RF: Software Fault Prediction Based on Robust Weight Based Optimization and Jacobian Adaptive Neural Network." *Concurrency and Computation: Practice & Experience* 34 (21): e7122.

"Iterated Feature Selection Algorithms with Layered Recurrent Neural Network for Software Fault Prediction." 2019. *Expert Systems with Applications* 122 (May): 27–42.

Jiang, Yue, Bojan Cukic, and Yan Ma. 2008. "Techniques for Evaluating Fault Prediction Models." *Empirical Software Engineering* 13 (5): 561–95.

Khalid, Aimen, Gran Badshah, Nasir Ayub, Muhammad Shiraz, and Mohamed Ghouse. 2023. "Software Defect Prediction Analysis Using Machine Learning Techniques." *Sustainability: Science Practice and Policy* 15 (6): 5517.

Li, Jian, Pinjia He, Jieming Zhu, and Michael R. Lyu. n.d. "Software Defect Prediction via Convolutional Neural Network." Accessed March 18, 2024. https://ieeexplore.ieee.org/abstract/document/8009936/.

Mehmood, Iqra, Sidra Shahid, Hameed Hussain, Inayat Khan, Shafiq Ahmad, Shahid Rahman, Najeeb Ullah, and Shamsul Huda. n.d. "A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning." Accessed March 18, 2024. https://ieeexplore.ieee.org/abstract/document/10155117/.

Rathore, Santosh S., and Sandeep Kumar. 2020. "An Empirical Study of Ensemble Techniques for Software Fault Prediction." *Applied Intelligence* 51 (6): 3615–44.

"Software Defect Prediction Based on Enhanced Metaheuristic Feature Selection Optimization and a Hybrid Deep Neural Network." 2021. *The Journal of Systems and Software* 180 (October): 111026.

"Software Fault Prediction: A Literature Review and Current Trends." 2011. *Expert Systems with Applications* 38 (4): 4626–36.

"Towards an Ensemble Based System for Predicting the Number of Software Faults." 2017. *Expert Systems with Applications* 82 (October): 357–82.

"Website." n.d. https://ieeexplore.ieee.org/abstract/document/9225352/.

TABLES AND FIGURES

Table 1. Pseudo code for Artificial neural network. The algorithm takes the dataset of software faults and helps to predict faults in it by using this algorithm.

Input: Software Faults prediction dataset
Output: Better Accuracy for Software Faults Prediction
Step 1: Start Step 2:Describe the artificial neural network requirements library. Step 3:Open a CSV file and load a dataset into it. Step 4:simultaneously encodes category features as part of the preprocessing step. Step 5:Separate the dataset into subgroups for testing and training. Step 6:Give the AI system knowledge to empower it. Step 7:Using an artificial neural network, make predictions. Step 8:Analyze the model's performance accuracy. Step 9:Lastly, subplots are created to show the two models next to each other. Step 10:In summary

Table 2. Pseudo code for Logistic Regressions. This algorithm takes the dataset of software faults and helps to predict it by using this algorithm.

Input: Software Faults prediction dataset
Output: Better Accuracy for Software Faults prediction
Step 1:start Step 2:Introduce yourself to the Logistic Regression library of prerequisites. Step 3:Put a dataset into a CSV file that has been opened. Step 4:encrypts category features in one go as part of the preprocessing stage. Step 5:A subgroup for testing and training should be created from the dataset. Step 6:Supply knowledge to the AI system Step 7:Make a prediction by employing Logistic Regression. Step 8:Assess the accuracy with which the model performs. Step 9:To display the two models side by side, subplots are finally created. Step 10:To sum up

Table 3. Improved accuracy of software fault prediction (Artificial Neural Network's accuracy of 84.10% and Logistic Regression accuracy of 61.90%)

Iteration Number	ANN Accuracy (%)	LR Accuracy (%)
1	84.30	61.90
2	84.40	61.50
3	83.00	61.40
4	83.20	60.60
5	83.50	60.20
6	83.10	60.00
7	82.60	59.90
8	82.20	59.40
9	81.70	61.30
10	81.40	60.80

Table 4. T-test with independent samples The confidence interval for the dataset was set to 95% while the level of significance was set to 5% (p>0.05) (with a p=0.151 value, Artificial neural networks performs better than Logistic Regression)

		Levene's Test for Equality of Variances		**T-test for Equality of Means**						
		F	**sig**	**t**	**df**	**sig(2-tailed)**	**Mean difference**	**Std.Error Difference**	**95% Confidence Interval of the difference**	
									Lower	**Upper**
Accuracy	Equal Variances Assumed	3.802	.067	38.58	18	.001	22.2000	0.57542	20.991	23.4089
	Equal variances not Assumed			38.58	15.49	.001	22.2000	0.57542	20.976	23.4231

Table 5. In a group statistical investigation, Artificial neural network and Logistic Regression were used. After 10 iterations, the mean accuracy value, standard deviation, and standard error mean for the Artificial neural network and Logistic Regression methods are obtained. The Artificial neural network approach outperformed the Logistic Regression method, according to the results.

Group	N	Mean	Standard Deviation	Standard Error Mean
ANN	10	84.10	1.52388	0.48189
LR	10	61.90	.99443	0.31447

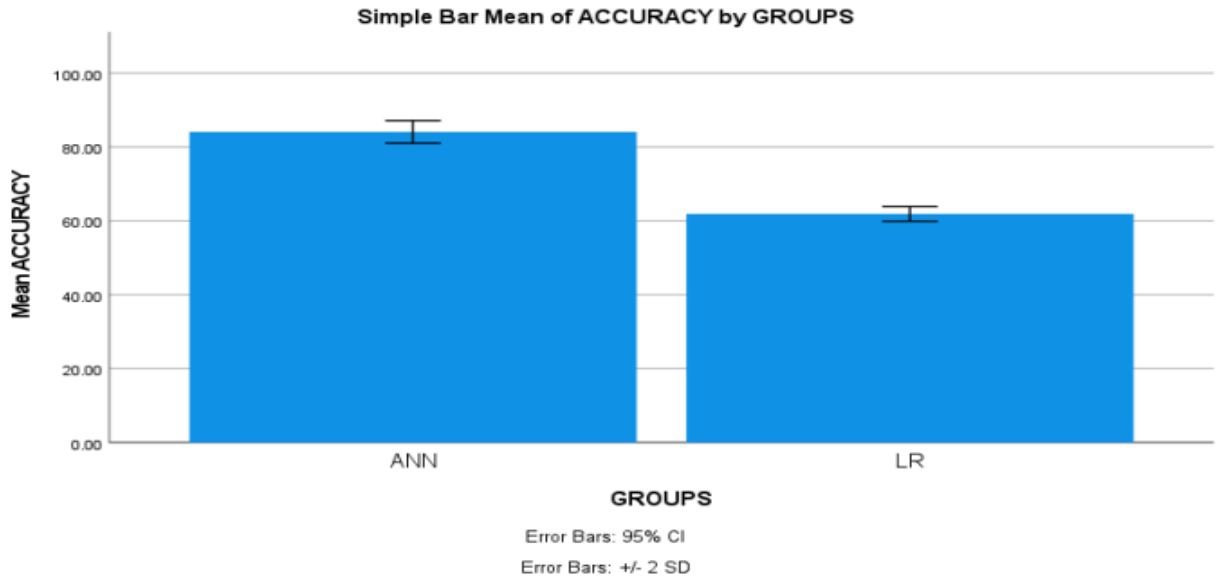

Fig.1 In terms of mean accuracy, Artificial neural network and Logistic Regression are compared. The mean accuracy of the Artificial neural network is higher than the Logistic Regression, while the standard deviation is slightly lower than the methods. X-Axis: Artificial neural network vs. Logistic Regression model; Y-Axis: Mean detection accuracy within +/-2 SD of 95% interval