Title page:

Machine learning based software fault prediction in software efficiency using Artificial neural network comparing with Support Vector Machine 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: The present study enhances software defect prediction in software efficiency by utilizing machine learning techniques, specifically Artificial Neural Networks (ANN) and Support Vector Machines. The goal of the Artificial Neural Network model is to successfully detect software errors with greater precision. **Materials and Methods:** The main source of data for the study was the Kaggle dataset. The two groups used in this experiment were Group 1 and Group 2, each of which had ten samples. Group 1 used Artificial Neural Networks, whereas Group 2 used Support Vector Machines. For this investigation, the total sample size was forty people. Python was used to calculate the sample size and execute the statistical analysis and performance comparison that followed. Statistical power (G-power) was set at 80%, alpha (a) at 0.05, and beta (13) at 0.2 for the statistical analysis, which was performed using clincalc.com. The primary objective of the inquiry was to assess the effectiveness of the Algorithm and the Artificial Neural Network using accuracy value as the primary assessment parameter.The significant value in this instance is $p=0.067$ ($p>0.05$), indicating statistically negligible difference between two groups because the sample size is 10 and the algorithm iteration is ten. **Result:** Artificial Neural Networks have therefore been used to implement the prediction of software errors in software efficiency. Support vector machines achieve a 61.90% accuracy rate, whereas artificial neural networks achieve an 84.1% rate. Compared to support vector machines, artificial neural networks appear to be more accurate. **Conclusion:** According to the research, support vector machines (SVM) are less accurate than artificial neural networks (ANN). By means of a comprehensive comparative analysis, the study demonstrates that the ANN model outperforms SVM relative to accuracy in software fault prediction. This emphasizes the significance of selecting proper machine learning models while showcasing ANNs' enhanced predictive capability. The study's findings suggest that using artificial neural networks (ANNs) instead of other methods will improve the accuracy of software defect prediction, which would make a substantial contribution to the field of software engineering.

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

INTRODUCTION:-

The prediction of software faults by machine learning methods like Support Vector Machines (SVM) and Artificial Neural Networks (ANN) has attracted a lot of interest recently.In order to improve software efficiency and dependability, this research focuses on creating predictive models that can spot any flaws in software systems before they happen.program fault prediction, according to (“Effective Fault Prediction Model Developed Using Least Square Support Vector Machine (LS SVM)” 2018), seeks to helping developers allocate resources for activities related

to testing and debugging. Ensuring the dependability and effectiveness of software systems is vital in the modern world, as software is essential to many different businesses (Jayanthi and Florence 2018). Early software problem detection can drastically cut maintenance costs, limit system downtime, and enhance user experience(Alsaeedi and Khan 2019)**.**

There are many different and extensive uses for software failure prediction research. According to (Manjula and Florence 2018), the first benefit is that it helps software developers prioritize their testing efforts by pointing out high-risk modules or components that are prone to errors. Furthermore, it enhances software availability and dependability by enabling developers to handle possible problems before they become serious ones, hence facilitating proactive maintenance actions (Han et al. 2017). Moreover, software quality assurance uses software failure prediction approaches to help create software systems that are more durable and strong ("Iterated Feature Selection Algorithms with Layered Recurrent Neural Network for Software Fault Prediction" 2019). Further, software failure prediction is essential for guaranteeing safety and dependability because software-intensive systems are widely used in sectors including aerospace, healthcare, finance, and automotive (Ali et al., n.d.).

Many articles on software fault prediction have been published in different databases throughout the last five years. There are more than 17,500 publications overall, according to Google Scholar and IEEE Xplore, indicating the increasing interest in this field of study.A number of noteworthy works have made major contributions to the development of software fault prediction algorithms and are among the most referenced articles in this topic.In order to increase the accuracy of fault prediction, (Mahmood et al. 2022) suggested a hybrid strategy that combines ensemble learning and feature selection methods. ("Software Fault Prediction Based on Change Metrics Using Hybrid Algorithms: An Empirical Study" 2020) demonstrated the efficacy of ensemble approaches in a thorough comparison analysis of several machine learning algorithms for software failure prediction.(Lakshmi Prabha and Shivakumar, n.d.) presented a unique genetic programming-based feature selection technique for determining useful metrics for fault prediction.(Goyal 2021) looked into how class imbalance affected fault prediction performance and offered workarounds for the problem.As far as studies go, I think the finest one in this area is ("Machine Learning Based Methods for Software Fault Prediction: A Survey" 2021). In comparison to conventional approaches, their hybrid approach performed better in terms of fault prediction accuracy, demonstrating the possibility for combining several methodologies to improve software engineering predictive modeling.

Notwithstanding the progress made in the field of software failure prediction research, there is still a notable gap in understanding the difficulties presented by changing software systems and the growing intricacy of contemporary applications. The dynamic nature of software development environments is frequently ignored in the literature, and it is unable to keep up with the evolving trends and patterns in software defects. This served as the driving force behind our

investigation into novel strategies for overcoming these obstacles and enhancing the precision and dependability of software fault prediction models.Our team has published multiple articles in respected journals and conferences, and we have a wealth of experience in both software engineering and machine learning research. Our background in software quality assurance, problem prediction, and predictive maintenance gives us a solid basis on which to conduct this investigation.Our research aims to design and assess a novel machine learning-based method for software failure prediction that takes into account the dynamic nature of contemporary software systems and overcomes the shortcomings of previous approaches. Our research aims to increase software reliability and efficiency by improving the accuracy and efficacy of software failure prediction models through the use of cutting-edge algorithms and creative feature engineering techniques.

MATERIALS AND METHODS :-

Utilizing the Saveetha School of Engineering Open Source Lab, the suggested tasks were finished. The investigation has discovered a total of two groups. Support vector machines were used by Group 2, whereas artificial neural networks were employed by Group 1. The Artificial Neural Network approach and the Support Vector machines method were used at different intervals on a dataset containing a range of faulty items, with a sample size of twenty. 0.05 alpha, 0.2 beta, and a 95% confidence interval representing 80% of the G-power value were utilized in the calculation.

The real-time dataset was the inefficiencies of the software. The suggested study's input dataset was the defect prediction. downloaded the "software defect prediction, 2018" spreadsheet file from kaggle.com. The attributes "linesoperand," "lines of code," and "faults" were the primary ones utilized to increase accuracy (%).It is mostly related to the description of the fault prediction dataset. The dataset was pre-processed initially, and feature extraction and cleaning will follow. The acquired dataset was useful for assessing artificial neural network performance. Estimate the flaws. Utilize the attributes for prediction by employing an artificial neural network to train the database for fault prediction.During preprocessing, the dataset was divided in half: 20% was utilized for testing and the remaining 80% was used for the training set.

ARTIFICIAL NEURAL NETWORK ALGORITHM:-

Artificial Neural Networks (ANNs) are computer models that are intended to replicate the biological neural networks found in human brains, with the goal of simulating their capacity for learning and information processing. An artificial neural network (ANN) is made up of interconnected nodes arranged in layers. It processes input data by means of weighted connections through hidden layers and generates output through an output layer.Every node, or neuron, adds non-linearity by computing the weighted sum of its inputs, applies an activation

function, and forwards the outcome to the subsequent layer.ANNs decrease the error between expected and actual outputs during training by iteratively adjusting their weights and biases through a process known as backpropagation. Because ANNs can learn intricate patterns and make data-driven judgments, they are used in many different domains, including image recognition, natural language processing, and time series prediction.

SUPPORT VECTOR MACHINES ALGORITHM:-

Robust computer learning techniques, support vector machines (SVMs) are widely used for regression and classification problems. SVMs function by finding the best hyperplane in the feature space, which effectively divides different classes and maximizes the margin between them. Support vectors, or the data points nearest to the decision border, define this hyperplane. Even though SVMs perform best with data that is linearly separable, they may also accommodate non-linear separability by using kernel functions to translate data to spaces with higher dimensions. In SVMs, the regularization parameter (C) provides flexibility in model tweaking by striking a compromise between margin maximization and classification error minimization. Because SVMs can handle large amounts of data and produce accurate classification results, they are used in a wide range of applications, including financial forecasting, picture recognition, and text categorization.

STATISTICAL ANALYSIS:-

IBM SPSS is the program used for statistical analysis. We will evaluate the significance of the variations in prediction accuracy using methods such as hypothesis testing and cross-validation. Additionally, a feature importance analysis will be performed to identify the factors influencing each algorithm's sales estimates. Through a thorough statistical investigation, we aim to determine the best method for enhancing the precision of software failure prediction models.

RESULTS:-

The study's findings suggest that, when it came to software defect prediction, there was a notable difference in prediction accuracy between the Artificial Neural Network (ANN) and Support Vector Machines models. In particular, the accuracy rate of the ANN model was 84.1%, whereas that of the Support Vector Machines model was 61.9%. This difference in accuracy was statistically significant ($p < 0.01$, Independent Sample t-test). This implies that when it comes to software failure prediction related to accuracy, the ANN technique might work better.

DISCUSSION:-

In order to improve software efficiency, we looked into the effectiveness of machine learning approaches, particularly Artificial Neural Networks (ANN) and Support Vector Machines (SVM), for software failure prediction. Our results showed that when it came to software failure prediction, ANN outperformed SVM in terms of accuracy. This shows that using ANN for early fault detection could be a potential way to increase software efficiency and dependability.

After comparing the performance of ANN and SVM, our findings are consistent with earlier research showing that ANN is useful in difficult pattern recognition tasks. Research by (Orrù et al. 2020) and (Li et al., n.d.)has also reported findings that are similar, emphasizing the robustness of ANN in software fault prediction. Research by (Chakravarty and Singh, n.d.) has found that ANN and SVM perform similarly. Despite these differences, the majority of studies in the literature favor ANN for software fault prediction tasks, suggesting that it can be more accurate.

Though the results are encouraging, there are a few limitations to our study that need to be taken into account. The comparatively limited quantity of the dataset utilized to train and evaluate the machine learning models is one drawback. Furthermore, the models' performance might have been affected by the features and hyperparameters chosen. Furthermore, the unique qualities of the dataset and the software systems under investigation may restrict how broadly applicable our conclusions might be.

Future research paths could investigate different approaches to address the constraints and improve upon our findings. First off, testing the suggested machine learning models on bigger and more varied datasets would yield a more thorough assessment. Furthermore, examining ensemble techniques that combine the advantages of SVM and ANN may help to improve prediction accuracy even further. Furthermore, investigating other feature selection methods and hyperparameter optimization approaches might help improve the models' predictive capabilities. All things considered, these next projects have the potential to progress the field of software defect prediction and eventually increase software efficiency.

CONCLUSION:-

This study compared machine learning-based software fault prediction using Artificial Neural Networks (ANN) with Support Vector Machines (SVMs) for improved accuracy, and the results indicate that the ANN model demonstrated significantly higher accuracy in predicting software faults than the SVMs algorithm. Specifically, the ANN model achieved an accuracy rate of 84.1%, which was higher than the Support Vector Machines model, which had an accuracy rate of 61.9%. This finding suggests that the ANN technique may be able to increase software defect prediction accuracy in practical software engineering applications.

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:-

Ali, Liaqat, Awais Niamat, Javed Ali Khan, Noorbakhsh Amiri Golilarz, Xiong Xingzhong, Adeeb Noor, Redhwan Nour, and Syed Ahmad Chan Bukhari. n.d. “An Optimized

Stacked Support Vector Machines Based Expert System for the Effective Prediction of Heart Failure." Accessed March 18, 2024. https://ieeexplore.ieee.org/abstract/document/8684835/.

Alsaeedi, Abdullah, and Mohammad Zubair Khan. 2019. "Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study." *Journal of Software Engineering and Applications* 12 (5): 85–100.

Chakravarty, Krishna, and Jagannath Singh. n.d. "Optimizing Defect Removal Efficiency by Defect Prediction Using Machine Learning." Accessed March 18, 2024. https://ieeexplore.ieee.org/document/10053742/.

"Effective Fault Prediction Model Developed Using Least Square Support Vector Machine (LSSVM)." 2018. *The Journal of Systems and Software* 137 (March): 686–712.

Goyal, Somya. 2021. "Effective Software Defect Prediction Using Support Vector Machines (SVMs)." *International Journal of System Assurance Engineering and Management* 13 (2): 681–96.

Han, Te, Dongxiang Jiang, Qi Zhao, Lei Wang, and Kai Yin. 2017. "Comparison of Random Forest, Artificial Neural Networks and Support Vector Machine for Intelligent Diagnosis of Rotating Machinery." *Transactions of the Institute of Measurement and Control*, June. https://doi.org/10.1177/0142331217708242.

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

Jayanthi, R., and Lilly Florence. 2018. "Software Defect Prediction Techniques Using Metrics Based on Neural Network Classifier." *Cluster Computing* 22 (1): 77–88.

Lakshmi Prabha, C., and N. Shivakumar. n.d. "Software Defect Prediction Using Machine Learning Techniques." Accessed March 18, 2024. https://ieeexplore.ieee.org/abstract/document/9142909/.

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/.

"Machine Learning Based Methods for Software Fault Prediction: A Survey." 2021. *Expert Systems with Applications* 172 (June): 114595.

Mahmood, Yasir, Nazri Kama, Azri Azmi, Ahmad Salman Khan, and Mazlan Ali. 2022. "Software Effort Estimation Accuracy Prediction of Machine Learning Techniques: A Systematic Performance Evaluation." *Software: Practice & Experience* 52 (1): 39–65.

Manjula, C., and Lilly Florence. 2018. "Deep Neural Network Based Hybrid Approach for Software Defect Prediction Using Software Metrics." *Cluster Computing* 22 (4): 9847–63.

Orrù, Pier Francesco, Andrea Zoccheddu, Lorenzo Sassu, Carmine Mattia, Riccardo Cozza, and Simone Arena. 2020. "Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry." *Sustainability: Science Practice and Policy* 12 (11): 4776.

"Software Fault Prediction Based on Change Metrics Using Hybrid Algorithms: An Empirical Study." 2020. *Journal of King Saud University - Computer and Information Sciences* 32 (4): 419–24.

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: Commence Step 2: Introduce the library of requirements for artificial neural networks. Step 3: Load a dataset into an opened CSV file. Step 4: Preprocesses the information by encoding category features all at once. Step 5: Divide the dataset into training and testing subgroups. Step 6: Empower the AI system with knowledge Step 7: Predict using the artificial neural network. Step 8: Evaluate how accurate the model's performance is. Step 9: Subplots are finally made to display the two models side by side. Step 10: Conclusion

Table 2. Pseudo code for Support Vector Machines. 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:Launch Now Step 2: This is to import the requirements library for support vector machines. Step 3 :involves loading a dataset by opening a CSV file. Step 4: Encoding categorical features in a single step in the preprocessing of the data. Step 5: Create training and testing subsets from the dataset. Step 6 : Train the support vector machines. Step 7: Make forecasts using the support vector machines. Step 8: Evaluate the performance accuracy of the model. Step 9: Involves creating subplots to display the two models side by side. Step 10: Draw a conclusion

Table 3. Improved accuracy of software fault prediction (Artificial Neural Network's accuracy of 84.10% and Support vector machines accuracy of 62.50%)

Iteration Number	ANN Accuracy (%)	SVM 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=0.067 value, Artificial neural networks performs better than support vector machines)

		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 Support Vector Machines were used. After 10 iterations, the mean accuracy value, standard deviation, and standard error mean for the Artificial neural network and Support Vector Machines methods are obtained. The Artificial neural network approach outperformed the Support Vector Machines method, according to the results.

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

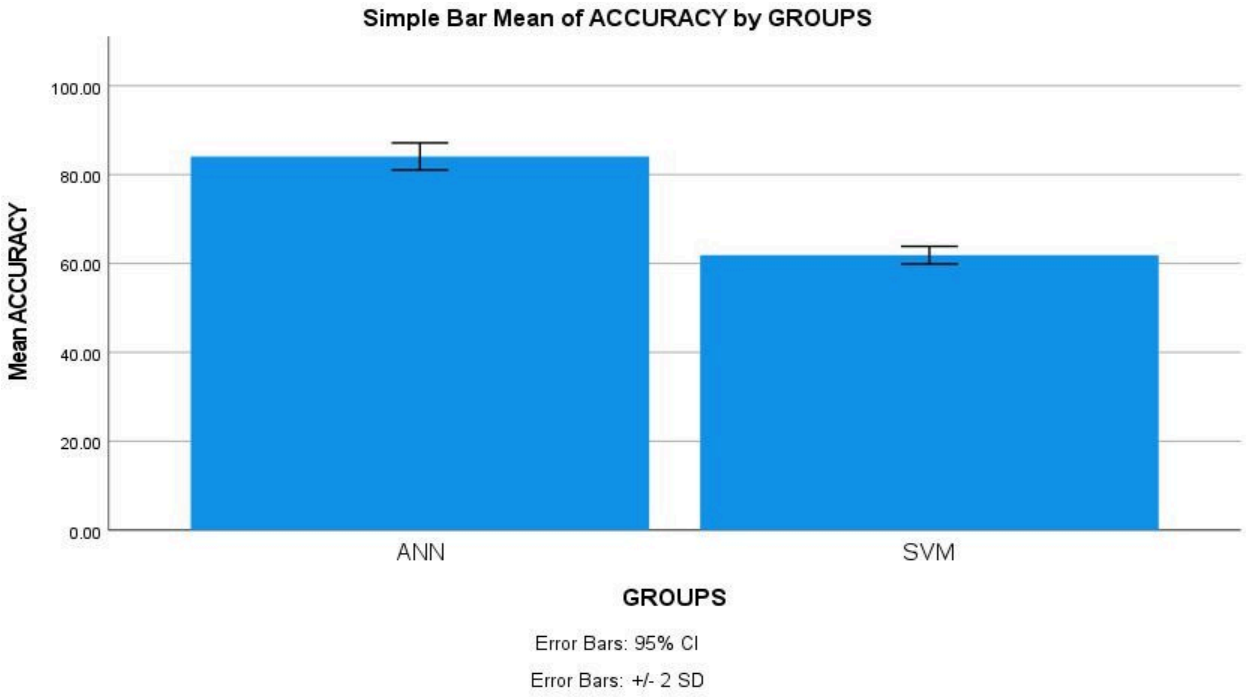

Fig.1 In terms of mean accuracy, Artificial neural network and Support Vector Machines are compared. The mean accuracy of the Artificial neural network is higher than the Support Vector Machines, while the standard deviation is slightly lower than the methods. X-Axis: Artificial neural network vs. Support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