EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

An empirically based object-oriented testing using Machine learning

Pusarla Sindhu [1,*], Giri Sainath Peruri [2] and Monisha Yalavarthi[3]

[1,2,3]Department of Computer Science and Engineering, GITAM University, Visakhapatnam-530045, India

Abstract

INTRODUCTION: The rapid growth of machine learning has the potential to revolutionize various industries and applications by automating complex tasks and enhancing efficiency. Effective software testing is crucial for ensuring software quality and minimizing resource expenses in software engineering. Machine learning techniques play a vital role in software testing by aiding in test case prioritization, predicting software defects, and analyzing test results.
OBJECTIVES: The primary objective of this study is to explore the use of machine learning algorithms for software defect prediction.
METHODS: Machine Learning models including Random Forest Classifier, Logistic Regression, K Nearest Neighbors, Gradient Boosting Classifiers, Catboost Classifier, and Convolutional Neural Networks have been employed for the study. The dataset includes a wide range of features relevant to software defect prediction and evaluates the performance of different prediction models. The study also focussed on developing hybrid models using stacking classifiers, which combine multiple individual models to improve accuracy.
RESULTS: The experimental results show that the hybrid models combining CatBoost and Convolutional Neural Network have outperformed individual models, achieving the highest accuracy of 89.5%, highlighting the effectiveness of combining machine learning algorithms for software defect prediction.
CONCLUSION: In conclusion, this study sheds light on the pivotal role of machine learning in enhancing software defect prediction.

Keywords: Software defect, Machine learning, Early defect prediction, Software quality assurance, stacking classifier

Received on 10 December 2023, accepted on 29 February 2024, published on 08 March 2024

doi: 10.4108/eetiot.5344

*Corresponding author. Email: spusarla@gitam.edu

1. Introduction

The field of machine learning is experiencing rapid growth. It has the potential to bring about a revolution in our daily lives and work processes by empowering intelligent systems to automate and enhance complicated tasks. Its applications span accuracy and efficiency, Personalization and customization, Predictive analytics, Fraud detection, Medical diagnosis and treatment, Natural Language Processing, and many other applications, allowing us to make informed choices and ameliorate complex tasks in various industries and applications. Software engineering has used it for various purposes, including defect prediction. The machine learning algorithm that produces the lowest error rates is considered the most effective for making predictions [1].

Humans have increasingly centered their attention on software-based systems over the past ten years, with software quality being the most important factor in user functionality. Software usage and size growth make it difficult to quickly identify structural flaws. These flaws could seriously harm the system, necessitating financial and human resources. Hence, effective software testing improves software system quality and lowers resource expenses and

damage costs. However, defect discovery may be challenging for software testers and developers in the early stages of the software development process. Each software company invests much money in software testing because it is crucial for the success of the software [2]. Having numerous software testing technologies handy, it is still hard to eradicate defects. So, effective bug identification is necessary, which doesn't sacrifice quality. Preventing errors reduces the burden of fixing them.

Technology like machine learning aids software engineering to enhance the testing procedure significantly. The technique establishes the corresponding prediction model by analyzing the software development process or code to predict errors. The outcomes can aid testers in identifying problematic modules or forecasting the number of problems in the modules in advance, aiding and directing decision-makers to utilize the limited testing resources [3].

Machine learning plays a crucial role in software testing by aiding in test case prioritization based on historical data, code changes, and defect patterns. It enables the creation of intelligent oracles by training models on past test data, allowing for detecting anomalies and deviations from expected behavior during testing. Machine learning algorithms can also predict software defects by considering code complexity, historical defect data, developer expertise, and project attributes. Additionally, machine learning assists in analyzing test results and logs to diagnose the root causes of failures, providing valuable insights for effective issue resolution.

Software quality assurance operations and activities can be utilized to detect software modules that have defects. This allows developers and testers to identify and address potential software issues, thereby facilitating the development of software systems of even better quality. Also, foresee whether or not the modules in the upcoming software version will have bugs. Identifying errors and fixing them are crucial steps in the software development process. An error is a mistake made by a human that can lead to flaws and bugs in the software. Software with errors does not satisfy the circumstances, requirements, and some user expectations [4]. When a software flaw is present and operational, it has the potential to induce system crashes or malfunctions, contingent upon the prevailing values of the variables.e

Software Defect Prediction (SDP) involves analyzing the software code and creating a predictive model that can classify defects into two groups: those likely to result in faults and those not [5]. The principal risks of failing to find software bugs in the early stages of software development are squandering time, resources, quality, money, and energy. These are faults that can occur in software at different stages. Software companies emphasize software quality more, particularly during the early stages of software development. One such approach is incorporating functional data and process artifacts into the source code, such as data that demonstrates the version control strategies employed by the developer for functional modifications. Integrating source code and software processing artifacts with software engineering enables more sophisticated automated learning and reasoning. The Software Defect Prediction (SDP) model is crucial for understanding, evaluating, and improving the quality of software systems because detecting errors early on can lead to efficient resource allocation, saving time and reducing the cost of developing high-quality software [6].

2. Literature Survey

Several studies have investigated software defect prediction using machine learning algorithms. Assim *et al.* proposed a method for predicting software defects based on machine-learning algorithms [1]. Tadapaneni *et al.* explored the application of machine learning and deep learning techniques for defect prediction [2]. Tian, Xiang *et al.* conducted comparative analyses of different machine learning-based defect prediction systems [3]. Cetiner *et al.* conducted a comprehensive study comparing various machine-learning approaches for software defect prediction [4].

Gururaj *et al.* examined various machine-learning algorithms used for defect prediction [5]. Razauddin *et al.* focused on efficiently predicting software defects using deep learning approaches [6]. Xu *et al.* proposed a GitHub-based data collection method for software defect prediction [7].

Krishnan *et al.* presented Activeclean, an interactive data-cleaning approach in the context of convex loss models [8]. Miao *et al.* compared machine learning models in text classification using Steam user reviews [9]. Mitchell *et al.* presented a parallel random forest classifier for R [10]. Andersen *et al.* applied logistic regression for modeling vibrotactile detection [11].

Bromley *et al.* introduced neural networks and k-nearest neighbor classifiers [12]. Fernández-Delgado *et al.* questioned the need for hundreds of classifiers in solving real-world classification problems [13]. Lee and Kim introduced a new ensemble learning technique with multiple stacking [15]. Joo *et al.* (2021) proposed an efficient healthcare service based on Stacking Ensemble [15]. Federmann *et al.* investigated using machine learning algorithms to improve phrase selection in hybrid machine translation [16]. Collectively, these papers contribute to the understanding and advancement of machine learning algorithms for software defect prediction. Ghosh et al.'s 2023 study [17] focuses on "Water Quality Assessment Through Predictive Machine Learning", highlighting the use of machine learning for analyzing and predicting water quality parameters.In "Unraveling the Heterogeneity of Lower-Grade[18] Gliomas," Rahat, Ghosh, and colleagues (2023) delve into deep learning-assisted segmentation and genomic analysis of brain MR images, offering new insights into this medical condition. Potato Leaf Disease [19] Recognition and Prediction using Convolutional Neural Networks," by Ghosh, Rahat, and team (2023), showcases the application of convolutional neural networks in accurately identifying diseases in potato leaves. Mandava,

Vinta, Ghosh, and Rahat's [20] 2023 research presents "An All-Inclusive Machine Learning and Deep Learning Method for Forecasting Cardiovascular Disease in Bangladeshi Population", integrating advanced AI techniques for health predictions. The 2023 study by Mandava et al., titled "Identification and Categorization of Yellow [21] Rust Infection in Wheat through Deep Learning Techniques", applies deep learning methods to detect and categorize wheat infections effectively. Khasim, Rahat, Ghosh, and colleagues' 2023 article, "Using Deep [22] Learning and Machine Learning: Real-Time Discernment and Diagnostics of Rice-Leaf Diseases in Bangladesh", explores AI-based solutions for diagnosing rice-leaf diseases. Deciphering Microorganisms through Intelligent Image Recognition", authored by Khasim, Ghosh, Rahat, and others in 2023, discusses [23] the use of machine learning and deep learning in identifying microorganisms through advanced image recognition techniques. The 2023 study by Mohanty, Ghosh, Rahat [24] and Reddy, "Advanced Deep Learning Models for Corn Leaf Disease Classification", focuses on the application of deep learning in classifying diseases in corn leaves based on a field study. Alenezi and team's 2021 research, "Block-Greedy and CNN Based Underwater Image Dehazing [25] for Novel Depth Estimation and Optimal Ambient Light", investigates novel CNN-based methods for enhancing underwater image clarity and depth estimation.

3. Data Elaboration

3.1 Data Collection

GHPR dataset encompassing a wide range of features relevant to software defect prediction was utilized to construct efficient prediction models provided by Jiaxi Xu[7]. The dataset consists of 6052 instances used for the baseline approaches. This dataset then undergoes data cleaning, exploratory data analysis, and proper outputs are generated.

3.2 Data Synthesis

The initial step involved addressing missing values to synthesize and clean the dataset. Next, the missing values in the dataset were replaced with the mean values of their respective columns, ensuring that the dataset remained complete and suitable for analysis.

Subsequently, the attention was directed to handling categorical data. Employing binary encoding techniques facilitated the conversion of the categorical variables into zeros and ones, effectively transforming them into a suitable format for further analysis.

Additionally, the data was cleaned thoroughly to ensure the dataset's integrity. This involved checking for inconsistencies, errors, and outliers in the data and resolving them appropriately.

The resulting dataset consisted of 6052 tuples with 23 columns, one of which serves as the target variable. Data synthesis and cleaning techniques enhanced the dataset's quality and suitability for more accurate and reliable analysis.

3.3 Development of prediction model

The data is divided into training and testing sets, with 80% allocated for training and 20% for testing. The Pandas library was used to scale the data frames to ensure that machine learning algorithms perform optimally by converting values to similar ranges. Experimentation is done over three hybrid models.

During training, the model adjusts its parameters to minimize the difference between its predictions and the actual target values in the training data. After the individual accuracies are obtained using the 10-fold cross-validation, the process for finding the accuracies for hybrid models has been considered.

The following section describes the models used for this study.

4. Analyzing Machine Learning Algorithms

4.1 Random Forest Classifier

A Supervised Machine Learning Algorithm widely used in Classification and Regression problems. This works on the bagging principle of the ensemble learning model, which chooses a random sample from the complete dataset, and individual decision trees are constructed for each sample, generating an output[10].

A boot-strapped aggregation method combines all the individual learners, and then the majority or the average of all the outputs is considered. It can handle large datasets with huge dimensionality. In the work presented, the parameters considered are n_estimators, which is the parameter that defines the count of trees in the model's forest. This parameter is set to 100, indicating that the random forest will generate 100 distinct decision trees. Random Forest Classifier attains an accuracy of 0.7010. This parameter governs the selection of samples and features, ensuring consistency in the results obtained from each iteration. The main reason to use this specific algorithm is that it prevents overfitting and provides high accuracy in the required time. These are the parameters defined for random forest in the study.

4.2 Logistic Regression

A supervised machine learning model and a statistical analysis method that gives a discrete output. It predicts the

categorical dependent variables, giving outputs such as True/False, 0/1, Yes/No, etc. It provides a probabilistic value of either 0 or 1 as the output[11]. These binary outcomes allow straightforward decisions between two alternatives. This easy-to-implement classification model achieves good performance in linearly separable classes. Classification algorithms are extensively used in industry. Logistic Regression uses a sigmoid function for predicting values to probabilities. The graphical representation would be an “S” shaped curve, as the value cannot exceed 0 and 1. Logistic regression has been implemented with the default parameters.

4.3 K Nearest Neighbors

The term "K nearest neighbor" refers to identifying the nearest neighbors for each sample. This approach involves multiple nearest neighbors being used to represent each sample. The fundamental concept underlying the KNN algorithm is that if most of the k-closest samples in a feature space are categorized in a specific category, a new sample will probably belong to that category and share similar attributes as the samples in that category[12].

KNN algorithm is used both for classification as well as regression. It uses all the available data for training without having a separate training phase. K is the number of nearest neighbors to be used in KNN. The features used in the KNN classifier for the study are n_neighbours = five and metric = ‘minkowski’.The n_neighbours parameter corresponds to the hyperparameter (k) for the k-Nearest Neighbors algorithm, while all remaining parameters are set to their default values. KNN does not make any assumptions about underlying data.

4.4 Gradient Boosting Classifier

In machine learning, Gradient Boosting is a highly competitive method, especially when there is limited training data, limited time for training, and limited expertise in parameter tuning[13]. The usage of decision trees makes the Gradient Boosting Machine a robust ensemble machine learning algorithm. Boosting, a typical ensemble technique, follows a sequential approach of adding models to the ensemble, with subsequent models aimed at improving the performance of prior models.

Each predictor in Gradient Boosting tries to reduce the errors of its predecessor. The feature applied in this study while using this algorithm is n_estimators, which is set to 100. Gradient Boosting, however, differs from other predictive methods in that it instead fits a predictor to the data as it moves up the gradient iteration. It fits a new predictor to the residual errors made by the previous predictor. CART(Classification and Regression Trees) is its base learner. This study has used both Gradient-boosting classifiers and regressors whose only difference is the loss function.

4.5 Catboost Classifier

CatBoost implements the gradient boosting algorithm that utilizes an ordered boosting approach. It is built upon the foundation of the gradient-boosting algorithm and employs oblivious decision trees as base predictors[14]. It automatically handles missing values and performs internal categorical feature encoding, simplifying data preparation. The default parameters have been used for building the model.

4.6 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a deep learning model designed explicitly for processing and analyzing visual data, such as images or videos. It consists of convolutional layers that learn and extract meaningful features from the input data, pooling layers that down-sample the feature maps, and fully connected layers for making predictions or performing classification tasks. CNNs have achieved remarkable success in various computer vision tasks, including image classification, object detection, and image segmentation, due to their ability to learn relevant features from raw pixel data automatically. They are widely used in applications where visual processing and analysis are crucial, making them popular in machine learning. Default parameters have been employed in this study.

4.7 Stacking

Stacking is an ensemble learning technique that merges the forecasts of multiple classifiers to establish a fresh training dataset for a meta-classifier. The process involves teaching multiple base classifiers with the full training set and using their predictions as inputs to train a meta-classifier. However, while stacking ensemble models can reduce model bias, they also risk overfitting the data because multiple models are trained on the same dataset[16]. Often termed a hybrid model, stacking demonstrates its prowess by fusing together an assortment of models to form a harmonized structure.

Typically, the base classifiers are termed level 1 models, and the meta-classifier is a level 2 model (or models). The architecture of the proposed stacking classifier is shown in Figure 1. Stacking combines multiple individual models to improve overall performance. The hybrid models combine a random forest classifier, logistic regression, and K-nearest neighbors for Model 01 and a random forest classifier, logistic regression, and gradient boosting classifier for Model 02. Both achieve higher accuracies when the stacking classifier is used as an ensemble model.

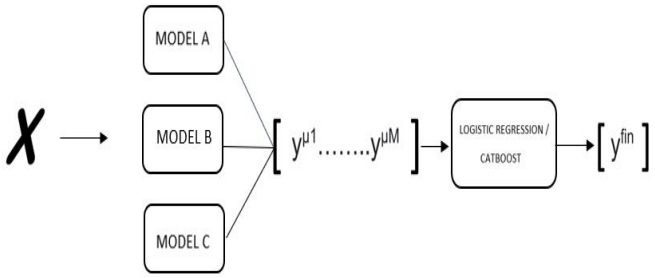

Figure 1. Architecture of the proposed stacking model

Table 1 displays various models used in stacking configurations and the corresponding meta-classifier for prediction. The meta classifier utilized throughout is Logistic Regression, except for the last configuration where the CatBoost Classifier serves as the meta classifier. This table provides a concise overview of the models employed and their associated meta-classifiers for stacking and prediction purposes.

Table 1. Models and Classifiers used in each of the stacking classifiers

Stack-ing	Model A	Model B	Model C	Meta Classifier
1	Random Forest	Logistic Regression		Logistic Regression
2	Random Forest	Logistic Regression	K Nearest Neighbors	Logistic Regression
3	Random Forest	Logistic Regression	Gradient Boosting Classifier	Logistic Regression
4	CatBoost Classifier	Convolut-ional Neural Network		CatBoost Classifier

5 Result Analysis

The study identifies the three models following an extensive experiment on various machine learning algorithms to determine the models with the highest accuracy when forming hybrid models, Hybrid approaches address common errors by effectively combining techniques from two or more algorithms. This combination is done in a way that optimizes the benefits of each algorithm.

The first stacking classifier implemented is between the random forest and logistic regression. The individual accuracies of random forest and logistic regression yield 0.7010 and 0.7229, respectively, depicted in Figure 2. The stacking of the two gives out an output of 0.7321, which has an increased accuracy.

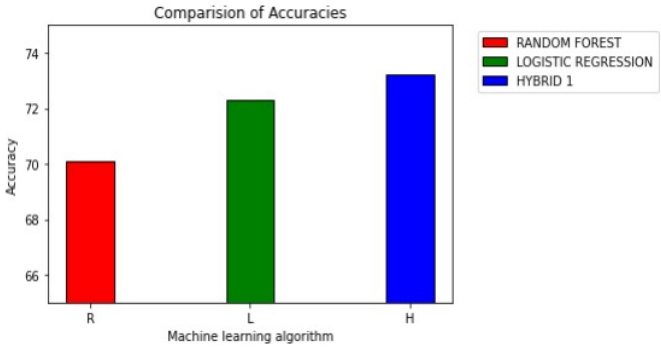

Figure 2. Comparison of accuracies with Hybrid 1

The second stacking classifier consists of random forest, logistic regression, and K nearest neighbor algorithms. These base learners are machine learning models with different strengths and weaknesses and perform poorly when used alone. The stack of these models resulted in accuracy going up to 0.7431 presented in Figure 3. This figure also illustrates how the accuracy of the second stacking classifier is superior to the accuracy of each base learner. The graph below compares each model's accuracy to the hybrid accuracy

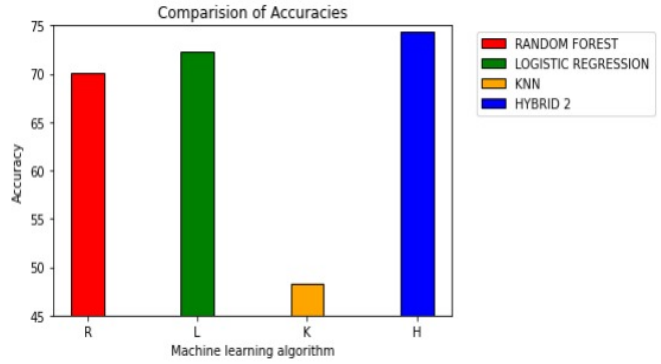

Figure 3. Comparison of accuracies with Hybrid 2

Figure 4 delineates the third stacking classifier, incorporating a combination of random forest, logistic regression, and gradient boosting models. The model's accuracy has significantly increased to 0.8008, showcasing the enhanced predictive performance achieved through this ensemble approach.

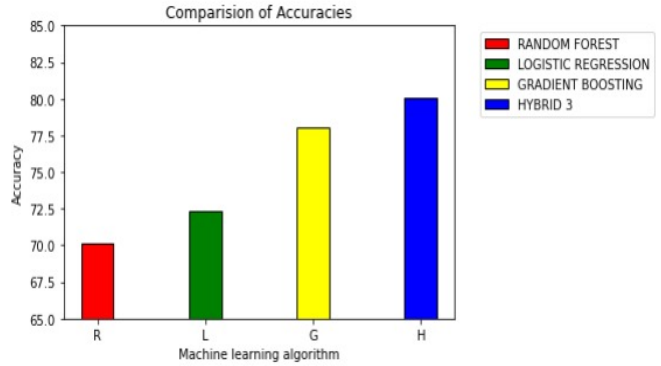

Figure 4. Comparison of accuracies with Hybrid 3

The fourth stacking classifier in Figure 5 consists of the CatBoost classifier and Convolutional Neural Network. This resulted in accuracy going up to 0.8955. This is a model of three machine-learning models which yielded lower accuracies individually. The graph below compares each model's accuracy to the hybrid accuracy.

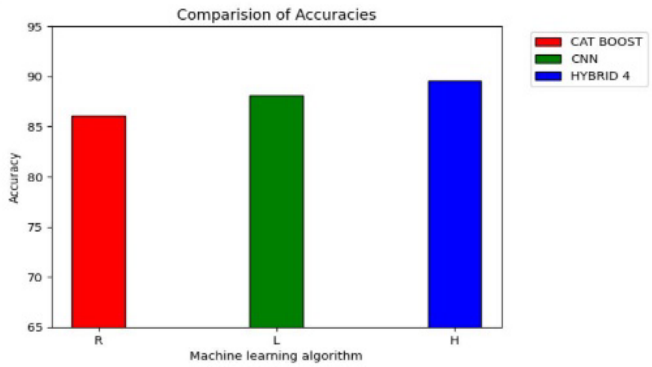

Figure 5. Comparison of accuracies with Hybrid 4

Comparing the models built, this study identified that the stack composed of Catboost Classifier and Convolutional Neural Network has the highest accuracy compared to the other hybrid models, achieving 0.895 accuracy. All these are implemented using the dataset mentioned in the work above.

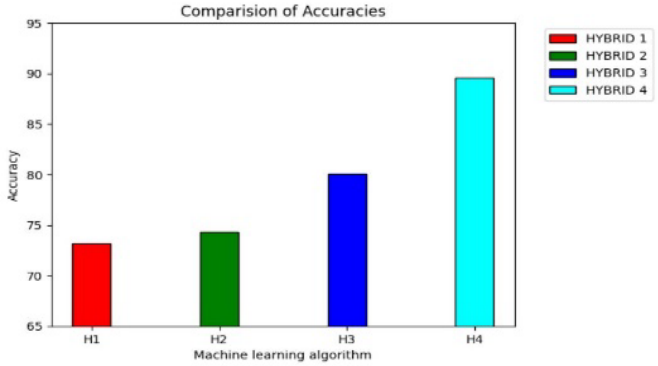

Figure 6. Comparison of accuracies of all the Hybrids

The training and testing accuracies of the individual models and the stacking classifier applied to the models are represented in Table 2.

Table 2. Performance comparison between all the considered models

Model	**Training Accuracy**	**Testing Accuracy**
Random Forest	0.7010	0.7704
Logistic Regression	0.7229	0.6540
K Neighbours	0.4826	0.5439
Gradient Boosting Classifier	0.7806	0.7908
CatBoost Classifier	0.8523	0.8064
Convolutional Neural Network	0.8632	0.8812
Stack 1	0.7321	0.7649
Stack 2	0.7433	0.7666
Stack 3	0.8009	0.7872
Stack 4	0.8845	0.8955

6. Conclusion

Machine learning algorithms can potentially revolutionize the field of software testing, particularly in defect prediction. Effective software testing is essential for software quality, improving the system and reducing resource expenses and damage costs. Machine learning aids software engineering in enhancing the testing procedure, allowing for the identification of problematic modules or forecasting the number of problems in the modules in advance. Software Defect Prediction (SDP) is a critical step in identifying and addressing potential software issues, facilitating the development of software systems of better quality. Using machine learning algorithms such as Random Forest Classifier, Logistic Regression, Gradient Boosting, K Nearest Neighbors, CatBoost Classifier, and Convolutional Neural Network has been shown to improve the accuracy of software defect prediction. By integrating machine learning techniques into software engineering, the study can achieve more sophisticated automated learning and reasoning, leading to more efficient resource allocation, saving time, and reducing the cost of developing high-quality software.In the future, we would like to experiment on the deep learning techniques in the stacking classifier.

7. References

1. M. Assim, Q. Obeidat and M. Hammad, "Software Defects Prediction using Machine Learning Algorithms," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 2020, pp. 1-6, doi: 10.1109/ICDABI51230.2020.9325677.
2. P. Tadapaneni, N. C. Nadella, M. Divyanjali and Y. Sangeetha, "Software Defect Prediction based on Machine Learning and Deep Learning," 2022 International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 116-122, doi: 10.1109/ICICT54344.2022.9850643.
3. Z. Tian, J. Xiang, S. Zhenxiao, Z. Yi and Y. Yunqiang, "Software Defect Prediction based on Machine Learning Algorithms," 2019 IEEE 5th International Conference on

Computer and Communications (ICCC), Chengdu, China, 2019, pp. 520-525, doi: 10.1109/ICCC47050.2019.9064412.
4. M. Cetiner and O. K. Sahingoz, "A Comparative Analysis for Machine Learning based Software Defect Prediction Systems," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225352.
5. S. A. K, V. Gururaj, K. R. Umadi, M. Kumar, S. P. Shankar and D. Varadam, "Comprehensive Survey of different Machine Learning Algorithms used for Software Defect Prediction," 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 2022, pp. 425-430, doi: 10.1109/DASA54658.2022.9764982.
6. Razauddin, S. Madhuri G, A. Oberoi, A. Vats, A. Sivasangari and K. Siwach, "Research on Efficient Software Defect Prediction Using Deep Learning Approaches," 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2022, pp. 549-554, doi: 10.1109/ICTACS56270.2022.9988292.
7. J. Xu, L. Yan, F. Wang and J. Ai, "A GitHub-Based Data Collection Method for Software Defect Prediction," 2019 6th International Conference on Dependable Systems and Their Applications (DSA), Harbin, China, 2020, pp. 100-108, doi: 10.1109/DSA.2019.00020.
8. S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg. Activeclean: Interactive data cleaning while learning convex loss models. In Arxiv: http:// arxiv.org/ pdf/ 1601.03797.pdf , 2015.
9. Youchen Miao, Zeyu Jin, Yumeng Zhang, Yuchen Chen, and Junren Lai. 2022. Compare Machine Learning Models in Text Classification Using Steam User Reviews. In 2021 3rd International Conference on Software Engineering and Development (ICSED) (ICSED 2021). Association for Computing Machinery, New York, NY, USA, 40–45. https://doi.org/10.1145/3507473.3507480
10. Lawrence Mitchell, Terence M. Sloan, Muriel Mewissen, Peter Ghazal, Thorsten Forster, Michal Piotrowski, and Arthur S. Trew. 2011. A parallel random forest classifier for R. In Proceedings of the second international workshop on Emerging computational methods for the life sciences (ECMLS '11). Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/1996023.1996024
11. Hans Jørgen Andersen, Ann Morrison, and Lars Knudsen. 2012. Modeling vibrotactile detection by logistic regression. In Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design (NordiCHI '12). Association forComputing Machinery, New York, NY, USA, 500–503. https://doi.org/10.1145/2399016.2399092
12. Bromley and Säckinger. Neural-network and K-nearestneighbor classifiers. Technical Report 11359-910819-16TM AT&T 1991
13. Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res, 15(1):3133–3181, 2014.
14. Yunsang Joo, Seungwon Lee, Hyoungju Kim, Pankoo Kim, Seongoun Hwang, and Chang Choi. 2021. Efficient healthcare service based on Stacking Ensemble. In Proceedings of the 2020 ACM International Conference on Intelligent Computing and its Emerging Applications (ACM ICEA '20). Association for Computing Machinery, New York, NY, USA, Article 28, 1–5. https://doi.org/10.1145/3440943.3444727
15. Lee Soo-eun, Kim Han-joon. (2020). A new ensemble learning technique with multiple stacking. Journal of the Korea Electronic Trade Association, 25(3) and 1-13.
16. Christian Federmann. 2012. Can machine learning algorithms improve phrase selection in hybrid machine translation? In Proceedings of the Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra) (EACL 2012). Association for Computational Linguistics, USA, 113–118.
17. Ghosh, H., Tusher, M.A., Rahat, I.S., Khasim, S., Mohanty, S.N. (2023). Water Quality Assessment Through Predictive Machine Learning. In: Intelligent Computing and Networking. IC-ICN 2023. Lecture Notes in Networks and Systems, vol 699. Springer, Singapore. https://doi.org/10.1007/978-981-99-3177-4_6
18. Rahat IS, Ghosh H, Shaik K, Khasim S, Rajaram G. Unraveling the Heterogeneity of Lower-Grade Gliomas: Deep Learning-Assisted Flair Segmentation and Genomic Analysis of Brain MR Images. EAI Endorsed Trans Perv Health Tech [Internet]. 2023 Sep. 29 [cited 2023 Oct. 2];9. https://doi.org/10.4108/eetpht.9.4016
19. Ghosh H, Rahat IS, Shaik K, Khasim S, Yesubabu M. Potato Leaf Disease Recognition and Prediction using Convolutional Neural Networks. EAI Endorsed Scal Inf Syst [Internet]. 2023 Sep. 21 https://doi.org/10.4108/eetsis.3937
20. Mandava, S. R. Vinta, H. Ghosh, and I. S. Rahat, “An All-Inclusive Machine Learning and Deep Learning Method for Forecasting Cardiovascular Disease in Bangladeshi Population”, EAI Endorsed Trans Perv Health Tech, vol. 9, Oct. 2023. https://doi.org/10.4108/eetpht.9.4052
21. Mandava, M.; Vinta, S. R.; Ghosh, H.; Rahat, I. S. Identification and Categorization of Yellow Rust Infection in Wheat through Deep Learning Techniques. EAI Endorsed Trans IoT 2023, 10. https://doi.org/10.4108/eetiot.4603
22. Khasim, I. S. Rahat, H. Ghosh, K. Shaik, and S. K. Panda, “Using Deep Learning and Machine Learning: Real-Time Discernment and Diagnostics of Rice-Leaf Diseases in Bangladesh”, EAI Endorsed Trans IoT, vol. 10, Dec. 2023 https://doi.org/10.4108/eetiot.4579
23. Khasim, H. Ghosh, I. S. Rahat, K. Shaik, and M. Yesubabu, “Deciphering Microorganisms through Intelligent Image Recognition: Machine Learning and Deep Learning Approaches, Challenges, and Advancements”, EAI Endorsed Trans IoT, vol. 10, Nov. 2023. https://doi.org/10.4108/eetiot.4484
24. Mohanty, S.N.; Ghosh, H.; Rahat, I.S.; Reddy, C.V.R. Advanced Deep Learning Models for Corn Leaf Disease Classification: A Field Study in Bangladesh. Eng. Proc. 2023, 59, 69. https://doi.org/10.3390/engproc2023059069
25. Alenezi, F.; Armghan, A.; Mohanty, S.N.; Jhaveri, R.H.; Tiwari, P. Block-Greedy and CNN Based Underwater Image Dehazing for Novel Depth Estimation and Optimal Ambient Light. Water 2021, 13, 3470. https://doi.org/10.3390/w13233470