Sindh Journal of Headways in Software Volume 2, Issue 1

Published Online 07-October-2023

Performance Analysis of Classification Algorithms for Software Defects Prediction by Mathematical Modelling & Simulations

Shadab Yameen Shaikh
Institute of Mathematics and Computer Science,
University of Sindh, Jamshoro, 76080, Sindh, Pakistan.
Email: shadabyameenshaikh@gmail.com

Naseem Afzal Qureshi
Department of Computer Science, Faculty of Science,
University of Karachi, Karachi, 75270, Sindh, Pakistan.
Email: qureshinaseemafzal@gmail.com

Muhammad Zohaib Khan
Shaheed Mohtarma Benazir Bhutto Institute of Trauma (SMBBIT), Karachi, Sindh, Pakistan.
Email: zohaib_khan2017@yahoo.com

Muhammad Ali Khan
Industrial Engineering and Management,
Mehran University of Engineering & Technology, Jamshoro, 76062, Sindh, Pakistan.
Email: muhammad.nagar@faculty.muet.edu.pk

Aisha Imroz
Avanza (Pvt.) Ltd, Karachi, Sindh, Pakistan.
Email: aishaimroz@gmail.com

Muhammad Ahmed Kalwar
Shafi (Pvt.) Limited Company, Lahore, Punjab, Pakistan.
Email: kalwar.muhammad.ahmed@gmail.com

Received: 18th March 2023; Accepted: 17th August 2023; Published: 07th October 2023

Abstract: This study explores machine learning (ML) techniques for Software defects prediction (SDP) by using Mathematical Modelling & Simulation. The SDP is also used in the critical systems of aviation, healthcare, manufacturing, and robotics. Many organizations face difficulty in forecasting the accurate defect before software deployment which is actually very crucial for estimating delivery time, maintenance efforts, and ensuring quality expectations. SDP enhances software quality by spotting potential defects in the upkeep phase. The current models of SDP rely on static program metrics for machine learning classifiers, but manual feature engineering may miss vital information impacting defect prediction accuracy. This study initially explores the past SDP results then aims to develop methods by adapting to future anomaly detection techniques. The study explores the various approaches of SDP which include K-Means methodology, Support Vector Machines (SVM) linear, Random Forest (RF) & Multi-layer Perceptron (MLP) algorithms and discussed the current models of SDP. The proposed SDP models are rigorously evaluated by using metrics like false alarm rate, precision, and detection rate. The results show high accuracy for K-Means and MLP (99.67%), K-Means and SVML (99.19%), and K-Means and RF (97.76%) for defect prediction.

Index Terms: Software defects prediction, Mathematical Modelling, Simulation, Machine Learning, Deep Learning, Artificial Intelligence, Performance analysis.

1. INTRODUCTION

Software defects prediction (SDP) is a critical area of research, focusing on identifying flaws in software applications and proposing innovative methods to address them. As software systems grow in complexity, the need for maintainable, high-quality, and cost-effective software becomes increasingly vital [1-3]. Early detection of flaws is essential to facilitate prompt rectification, leading to improved software reliability and performance [4]. Manual code reviews are time-consuming and impractical for large codebases, making automated SDP algorithms crucial to manage finite resources effectively [5-6]. Over the past three decades, software defect prediction has seen significant advancements, with various approaches classifying software components as defect-prone or non-defect-prone, identifying defect associations, and estimating remaining faults in software systems. This research focuses on developing software defect prediction models based on past failure data and software parameters to classify modules and classes accordingly [7-8]. By concentrating testing resources on error-prone areas, developers can achieve higher product quality within project timelines and budgets[9-11].Defect identification, analysis & reduction is critical to improve organizational performance [12-14]. It contribute towards improved organizational excellence [15]. Defect reduction improve customer retention in service organizations [16]. Software's with reduces/zero defects can improve information retrieval & knowledge management [17]. The employees of software development organizations of Pakistan also face the tremendous work stress [18]. Modern and updated ICT applications also contribute in the reduction of software defects [19-21]. Learning organizations have the proven records of performance improvement in organizational operations by the implementation of quality software applications, AI & ML techniques [22-28]. Many previous studies on SDP focused the susceptibility of software components by analysing metrics obtained from the code [29]. Despite various attempts to utilise machine learning techniques, none of the methods have demonstrated consistent reliability. Many organizations in Pakistan acknowledge the applications AI & ML software's in the optimization of operations but still lag behind [30-31]. The recent applied case studies of Pakistani organizations in the context of optimization by better quality software applications include procurement report [32], routine report making [33], purchase order [34], acquisition report [32], planning report [35], Supplier Price Evaluation Report [36], material delivery time analysis [37], product mix & profit maximization [38], order costing analysis [39], production plan [40], demand management [41], procurement report [34] and material cost comparative analysis [42]. Whereas the recent applications of Pakistani hospitals in the context of optimization by better quality software applications include hospitals' outpatient departments [43-48] and emergency Health Care Units of Pakistan [49-50].This study employs supervised and unsupervised learning techniques for software defect prediction, using K-Means clustering and Support Vector Machines Linear, RF, and MLP algorithms for clustering, LR, and classification purposes. These techniques exhibit enhanced recall, accuracy, f1-score, prediction, precision, clusters, and classifiers, promising improved defect prediction accuracy.

2. LITERATURE REVIEW

Performance Analysis of Software Defects Prediction is the area of concern for cyber security professionals due to security threats and increasing phishing attacks [51-54]. Mathematical modelling, simulations, IoT, AI and ML are being used effectively to evaluate the performance of SDP [55-59]. DL and Industry 4.0 are also the recent developments in the techniques to improve the Cyber security and to safeguard the organizations' critical systems from phishing attacks [60-65]. Performance Analysis of software has been performed by many experts with various Mathematical modelling &simulations techniques [66-69]. Medical field is getting the remarkable results by using the machine learning techniques for the more accurate diagnosis & prediction of diseases at the individual and public level [70-75]. The systematic review of SDP models was performed by many researchers and the results of various models were compared [76-79]. The SDP models with ML & empirical assessment were critically evaluated by the researchers and proposed frameworks were developed by them for better results of Software Defects Prediction[9], [80]–[82]. Simulation can be used as an effective for SDP [83-85]. Numerous projects have been successfully implemented SDP by simulation tools & techniques [86-88]. Propagation neural network model, poisson regression, spiderhunt-based deep convolutional neural network classifier and discrete mycorrhiza optimization nature-inspired algorithm are used effectively researchers for SDP [89-92]. Hassan et al. achieved more than 99% accuracy on the dataset with an integrated approach for sentiment classification and information retrieval techniques [93]. Mathematical Modelling & Simulation is getting popularity for the prediction of software defects. The ROCUS, Ayesian networks, Petri nets, AHP and boosting approach are amongst the effective Mathematical Modelling & Simulation techniques for SDP[94], [97-98]. Machine Learning is also getting popularity for predicting software defects and researchers consider it as effective techniques [99-102]. The recently completed software prediction projects are the quite evident of the fact that machine learning also proved its worth in the field of SDP [103-107].Deep Learning is an effective AI based tool for predicting software defects [108-110]. There are very few recently completed projects of software defects prediction projects by using deep learning technique but they have shown the remarkable results [111-115]. SVM is a type of supervised learning algorithm which is comparatively new machine learning tool in the field of SDP to solve classification problems [116-119]. Though there are very few recently completed projects of software defects prediction projects by using support vector machine technique but they have proved the effectiveness in SDP [120-121].K-means clustering can be used effectively to increase software defect prediction [122-124]. Researchers quoted the benefits & applications of K-means in the various fields to predict the software defects [125-128].Practitioners used Random Forest in SDP projects and mentioned its benefits [129-135]. A multilayer perceptron (MLP) is a misnomer for a feedforward artificial neural network, consisting of fully connected neurons with a nonlinear activation [136-138]. The recently completed software prediction projects are the quite evident of the fact that MLP also proved its effectiveness in the field of SDP [139-142].

3. PROBLEM STATEMENT

There is the growing need of more accurate Software defects prediction (SDP) from modern complex systems to daily routine systems. SDP is also used in the critical systems of aviation, healthcare, manufacturing, and robotics where the prediction of accurate defect before software deployment is actually very crucial for estimating delivery time, maintenance efforts, and ensuring quality expectations. Despite many developments still many organizations face difficulty in forecasting the accurate defect before software deployment. SDP enhances software quality by spotting potential defects in the upkeep phase. Several Mathematical Modelling, Simulation, Artificial Intelligence (AI) & Machine Learning (ML) techniques are in discussion for SDP. The current models of SDP rely on static program metrics for machine learning classifiers, but manual feature engineering may miss vital information impacting defect prediction accuracy. The objective of this study is to compare the previous models of SDP and their results then aims to develop methods by adapting to future anomaly detection techniques. To achieve this, it is crucial to explore various machine learning approaches and prediction models that can accurately predict software defects outcomes using the available dataset. The performance of these models needs to be evaluated and measured. This research aims to address these challenges by utilizing the selected dataset and analyzing the performance of different machine learning algorithms in developing prediction models. This paper is divided into four (4) sections. The first section provides an introduction to the research study. Section 2 discusses the related work in the field of research. Section 3 focuses on the results and discussion of various algorithm combinations used for predicting the software defects. Lastly, the concluding section presents a statement on the most efficient algorithm combination.

4. BACKGROUND

4.1 Classification, Regression, And Clustering in Machine Learning

In machine learning, classification involves categorising different federation mechanisms into discrete groups and subclasses based on their similarities. The systematic method of dividing systems into recognizable groupings and subcategories depending on their commonalities is called classification. Many researchers used the concepts of classification, regression and clustering in Machine Learning to analyse & investigate the diseases [143-147]. Linear regression, linear classification, and Naive Bayes classifier are three common methods of categorization. Classifications are typically applied to organised and labelled data. Figure 1 shows a range of classification techniques used in various operations.

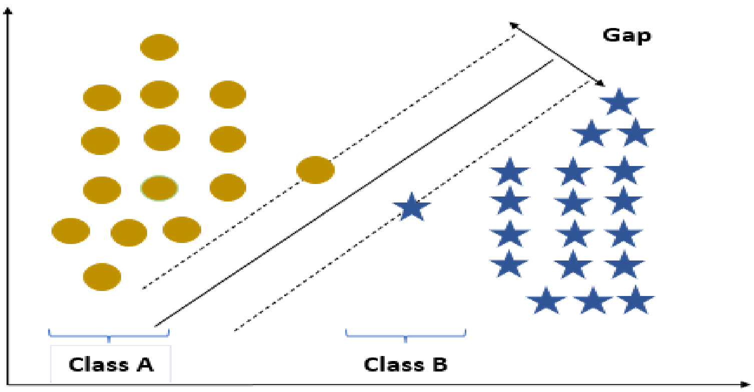


Figure 1: Overview of Classification [148]

Linear and nonlinear regression models require different types of supervised and unsupervised learning methods due to the diverse nature of the interactions between independent and dependent variables in each model. These approaches are utilised to perform regression tasks.

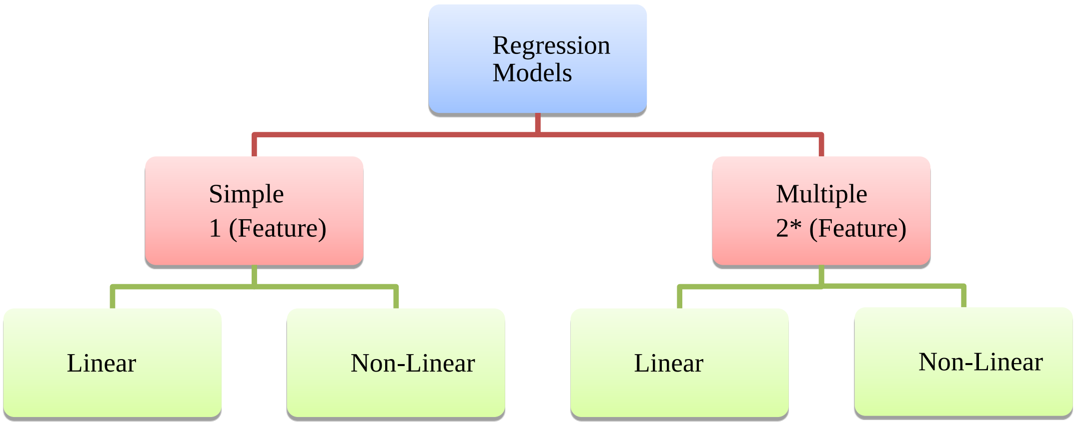


Figure 2: Regression Models

Figure 2 depicts how machine learning algorithms utilize a range of regression properties, both unstructured and structured data. Machine learning techniques employ both organized and unstructured data, as well as a variety of regression features. Both of these non-linear and linear regression incorporates the first and second properties of the regression model. Clustering is a particularly common kind of learning that is unsupervised, which has many uses across several sectors. A cluster is a group of related pieces of information that have undergone isolation and processing based on a data machine (ID).Figure 3 depicts numerous clusters of diverse things.

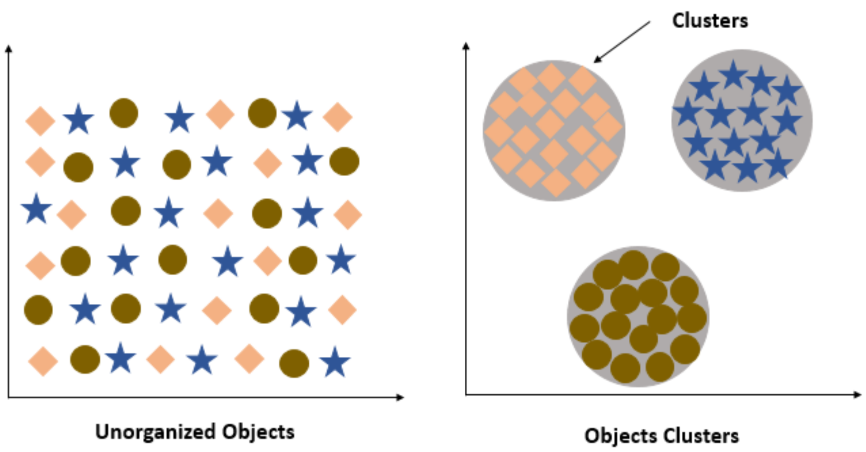


Figure3: Clustering [149]

5. PROPOSED METHODOLOGY

This portion provides an overview of the process for developing a work breakdown structure for software defects prediction (SDP).

1. The first step involves retrieving the dataset from Google Drive.
2. Next, the data undergoes various procedures such as data cleaning, feature extraction utilising methods like (CountVectorizerandTfidfTransformer), pre-processing, and standardisation using (MinMaxScaler).
3. Standardisation requires the creation of a system to transform variable frequency and amplitude, such as (0.98671539), and performing a standardisation analysis to acquire the output.

4. The K-Means clustering unsupervised machine learning technique is subsequently employed to enhance the precision, recall, f1-score and accuracy of the model.
5. The data is then split into train and test data sets, with the train data size set at 0.75 percent and the test data size set at 0.25 percent, to implement this technique.
6. Finally, the SVML, RF, and MPL algorithms are used to construct the ultimate model.

Figure 4 illustrates the software defects prediction (SDP) architecture, providing a clear perspective on the research project and a brief summary of the work breakdown structure.

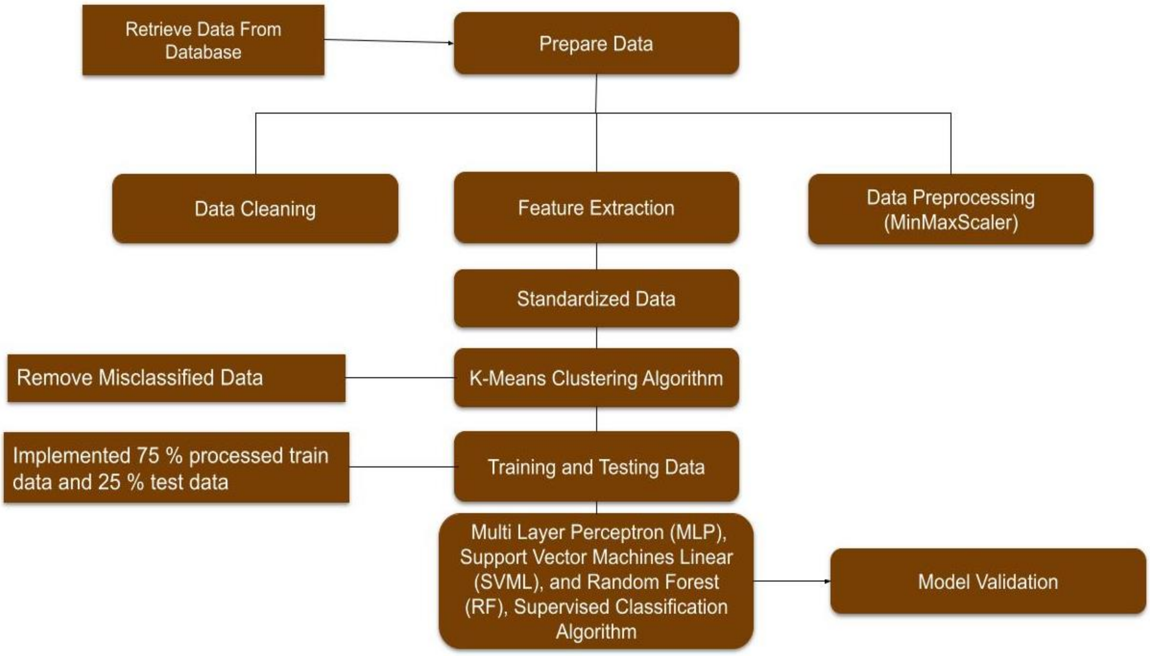


Figure 4 Proposed algorithm for Software Defects Prediction (SDP)

The flowchart depicts retrieving data from a database, pre-processing the data to normalise and standardise it using data cleaning methods, and then using clustering and classification techniques to implement the processed training data (75%) and test data (25%) for model validation. The algorithm is composed of two distinct sections: data pre-processing and classification.

5.1 Preprocessing

During the early processing stage, we sanitise the data and apply clustering techniques to extract relevant information. In order to achieve this, we explore two popular approaches, namely, K-Means clustering, which is explained below. Later, in the classification stage, we perform additional data manipulation on the processed data.

5.1.1 Performance Analysis

Python is a language primarily used for scripting, which finds wide application in various domains such as programming, machine learning, web development, and databases. In this study, the Anaconda Navigator ->Jupyter Notebook GUI framework is employed and Python is used to link datasets and implement various algorithms such as K-Means, Random Forest, Support Vector Machines Linear, Multi-layer Perceptron. Our dataset pertains to software defects prediction (SDP) and involves predicting whether a software contains defects or not based on software bugs. The dataset consists of 22 attributes or characteristics (columns) and 10,885 instances or observations (rows). We ran three separate programs using the same dataset. The first program utilised K-Means and Multi-layer Perceptron (MPL), the second program used K-Means and Support Vector Machines Linear (SVML), and the third program used K-Means and Random Forest (RF). All of these programs were executed on a personal computer with the following configuration:

- The computer is equipped with an Intel Core (TM) i5-2520M (2nd Generation) CPU operating at 2.50 Gigahertz.
- It has a RAM capacity of 4 GB.
- It is running on a 64-bit OS, specifically Windows 10 (Home).
- It has a 500 GB hard disk.

5.1.2 Data Collection

We obtained the Software Defects Prediction (SDP) dataset from Kaggle, which is a platform hosting various machine learning datasets. Ihsan & Aquil previously used this particular dataset in their research [150]. It comprises 10,885 instances or observations, each with 22 attributes representing the specifications of software applications and their measures related to SDP. The target class in this dataset represents the status of each outcome, with a total of 5,427 not-defects software bugs and 5,458 defects software bugs [151]. Table 1 presents a concise summary of the parameters and features that are included in the SDP dataset utilised in this research for the purpose of forecasting software defects.

Table 1 Original Dataset Used for Predicting Software Defects

Parameters of the dataset	**Characteristics of SDP**
loc	count of program statements
v(g)	complexity of cyclomatic
ev(g)	Intrinsic complexity
iv(g)	Complexity of the design
n	count of operands and operators
v	Amount of space
l	Length of the program
d	adversity
i	Intellect
e	Exertion
b	no of errors
T	Time predictor
lOCode	count of lines
lOComment	total comment lines
lOBlank	total whitespace lines
lOCodeAndComment	Count of lines with code and comments
Uniq_Op Unique	distinct Operators
Uniq_Opnd Unique	distinct Operands
Total_Op	Overall operator count

Total_Opnd	Overall operator count
branchCount	branch count of flowchart
defects	defects reported

The goal of this project is to investigate the necessary steps for predicting software defects, including data normalisation, pre-processing, simulation, and induction requirements. Other aspects such as critical criteria, complexity issues, post-processing, and system effectiveness are also examined. The first step is to gather facts from the dataset, followed by preparing and pre-processing the data, including normalisation and standardisation. Table 2 presents the resulting cleaned and pre-processed dataset. Additionally, Figure 5 provides a visual representation of complex information without K-Means execution. The X value is represented by a purple colour circle, and the Y value is represented by the yellow colour circle.

Table 2 Dataset for predicting software defects, which has been processed

```
22-Dimension

array ([[0.36223789, 0.60325949, 0.25972736, ..., 0.04290384, 0.99847326, 0.79664566],
    [0.20296517, 0.47553557, 0.51124005, ..., 0.01224384, 0.39541578, 0.66811618],
    [0.17949324, 0.12738392, 0.65493002, ..., 0.35573798, 0.03057093, 0.34464949], ...,
    [0.9456746, 0.98671539, 0.38383904, ..., 0.52999682, 0.31716936, 0.70528904],
    [0.13678812, 0.82731781, 0.71771077, ..., 0.02882109, 0.29340566, 0.69901713],
    [0.69547178, 0.63604136, 0.42970602, ..., 0.64185376, 0.03466157, 0.37666046]])
```

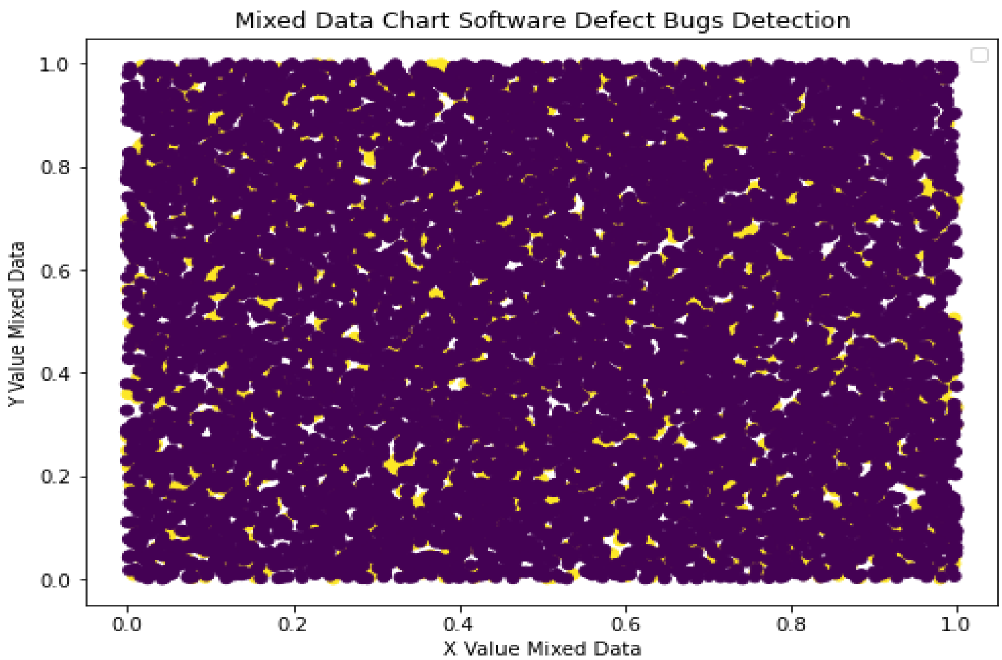


Figure 5: Mixed Data Chat Software Defects Prediction (SDP)

5.1.3 *Artificial Intelligence*

"Artificial intelligence (AI) is a branch of computer science that focuses on developing smart computers capable of performing tasks that typically require human intelligence. This field involves the creation of algorithms and models that enable computers to analyze data, make logical deductions, and generate predictions or conclusions" [76]. Artificial intelligence encompasses various domains such as robotics, machine learning, natural language processing, computer vision, and more. Its objective is to imitate and automate cognitive functions like decision-making, pattern recognition, and problem-solving.

5.1.4 K-Means Clustering Algorithm

The most popular kind of unsupervised learning, known as clustering, has a wide range of uses and widespread adoption in several fields. In order to create a set of data identified as clustering, information must be broken up and processed by a computer. Every cluster is assigned a distinctive identification number for identification purposes. The unsupervised K-means method is a machine learning technique that classifies data into two categories: unstructured and mixed. The dataset begins with a set of randomly selected average values that serve as the starting point for each subsequent group. The location of the intermediate values is then calculated to improve the clustering [152]. The fundamental principles that underpin the K-means algorithm are as follows:

1. Identify the most suitable number of clusters (K) for use in the clustering process
2. Sort the dataset and randomly select K values to be the centroids before calculating the centroids.
3. After the centroids no longer change, identify the clusters. However, the overall approach to clustering the data remains the same.
4. Calculate the number of patterned lengths between each centroid and the data points.
5. Allocate each data point to the cluster that is closest to it.
6. Calculate the sum of all data points assigned to each cluster to obtain the cluster centroids.
7. Complete the clustering process.

Several scientific methods and metrics, such as Euclidean, Manhattan, and Hamming measures, were employed to classify each program in the dataset.

Euclidean	$\sqrt{\sum_{i=1}^{k} \quad (xi - yi)2}$	Equation 1
Manhattan	$\sum_{i=1}^{k} \quad \lvert xi - yi \rvert$	Equation 2
Minkowski	$\left(\sum_{i=1}^{k} \quad (\lvert xi - yi \rvert)q\right)1/q$	Equation 3

In this processing, the standard collection is used to create mixed data representations through a pre-processing technique. K-Means was used to filter and process large datasets, making it easier to understand the data and remove redundant information. Through the utilisation of clustering, we were able to detect two distinct clusters and assign a likelihood score to each piece of information in order to determine its membership within a given cluster. This method resulted in a member matrix that shows the association between each sample and its respective cluster. The approach involves using a clustering methodology, such as the K-Means algorithm and centroid clustering values, and executing it on a 22-dimensional dataset with binary-class data. Each data point is associated with a centroid based on the distance between them. The closer the cluster is to the data centroid, the stronger the association. The SDP dataset is a 22-dimensional dataset that includes features related to software defects prediction values and an attribute that targets property cluster number. We have briefly discussed the K-Means Clustering centroid value and included Figures 6 and 7 to illustrate the clusters and the sum of squared error line charts for the 22-Dimensional binary-class datasets, respectively, after transforming unstructured material into structured data.

Table 3 K-Means Clustering Centroid Value

```
Array ([[0.49914726, 0.49098853, 0.5040306, 0.48515271, 0.51490666, 0.51906228, 0.47665096, 0.4984902, 0.49849351, 0.51083994, 0.50353293, 0.50095931, 0.50579481, 0.5030984, 0.49457925, 0.750223, 0.50110969, 0.49787315, 0.50347232, 0.49546553, 0.48247945, 0.48096822],

[0.50335415, 0.50101048, 0.48892057, 0.51358704, 0.48948397,0.48769329, 0.51316378, 0.50301923, 0.50445169, 0.49481033, 0.48963746, 0.49711861, 0.49109309, 0.49119229, 0.51433448, 0.25323482, 0.50181001, 0.50683171, 0.49787394, 0.50327462, 0.51617455, 0.51853753]])
```

Table 4 K-Means Two Clusters Pre-processed Software Defects Prediction (SDP) Dataset

```
array ([[0.36223789, 0.60325949, 0.25972736, ..., 0.04290384, 0.99847326, 0.79664566],
[0.20296517, 0.47553557, 0.51124005, ..., 0.01224384, 0.39541578, 0.66811618],
[0.17949324, 0.12738392, 0.65493002, ..., 0.35573798, 0.03057093, 0.34464949] ...,
[0.9456746, 0.98671539, 0.38383904, ..., 0.52999682, 0.31716936, 0.70528904],
[0.13678812, 0.82731781, 0.71771077, ..., 0.02882109, 0.29340566, 0.69901713],
[0.69547178, 0.63604136, 0.42970602, ..., 0.64185376, 0.03466157, 0.37666046]])
```

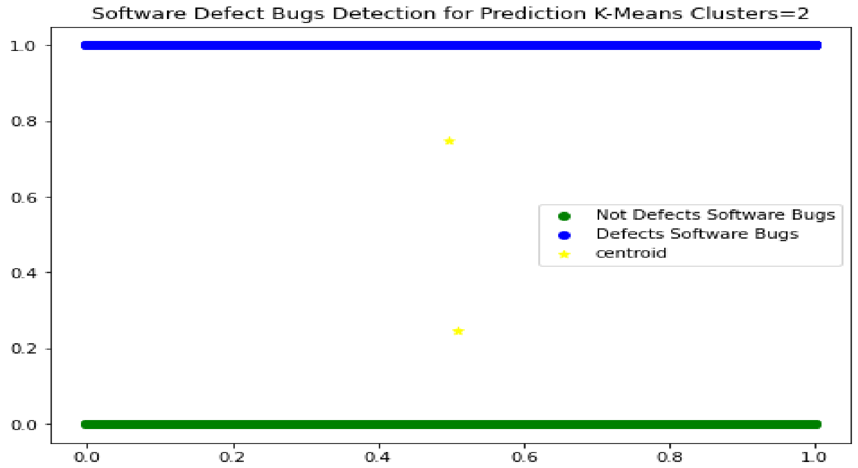


Figure 6: K-Means Two Clusters Software Defects Prediction (SDP)

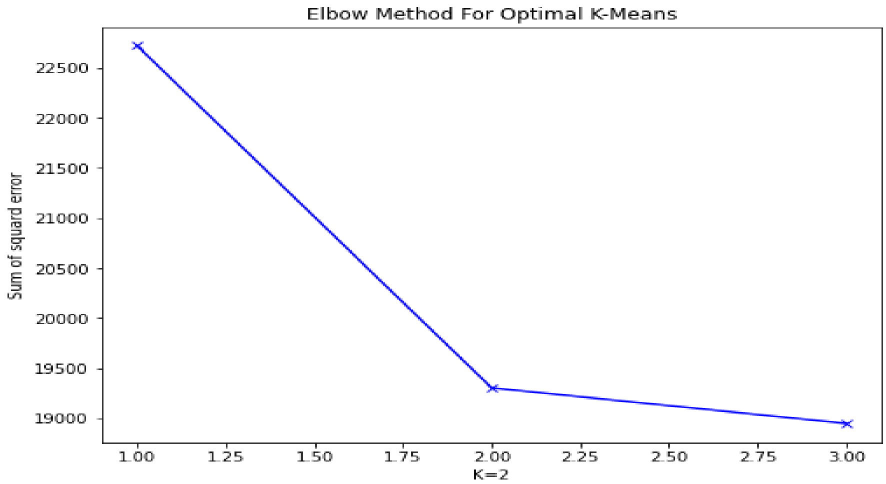


Figure 7: K-Means Sum of Squared Error Line Chart

To evaluate the effectiveness of CFD using both clustering methods, precision, recall, and f-measure are used. A concern score is determined by measuring how much the system deviates from the standard, and the result is classified as valid, suspicious, or illegal.

5.2 Classification

The Classification algorithm is a type of supervised learning that categorises observed data using training data. The process of grouping observed data into different categories or sections is called classification. To determine which classifier performs the best in our dataset, we test several different classifiers.

5.2.1.Multi-Layer Perceptron's (MLP) Algorithm

An advanced optimization algorithm called the Multilayer Perceptron (MLP) is composed of multiple perceptron's. MLP consists of an input layer that receives input data, an output layer that generates judgments or estimates based on the input, and an arbitrary number of hidden layers that serve as the MLP computational power. By varying the number of hidden layers, the MLP is capable of approximating any continuous function"[153], [154]. In cases where datasets are not conditionally independent, the MLP overcomes this challenge by employing participants to develop machine learning and prediction models with a more flexible and complex framework. This approach, often used in supervised learning, addresses challenges related to difficult data patterns and enables scientific advancements in various fields. Some of these approaches, such as Linear, Non-linear Regression, Sigmoid, and Cost Linear, are constructed based on the principles of classification.

Sigmoid	$S\,(z) = \frac{1}{(1+e^{-z})}$	Equation 4
Linear Regression	$y \;=\; e\hat{\ }(b0 \;+\; b1 * x) \,/\, (1 \;+\; e\hat{\ }(b0 \;+\; b1 * x))$	Equation 5
Cost Linear Regression	$(Cost\ (h\theta(x), y)) \;= -log\ (h\theta(x)), if\ y = 1\ and$	Equation 6
	$\big(Cost\ (h\theta(x), y)\big) = -\ log\ \big(1 - h\theta(x)\big), if\ y = 0$	
Nonlinear Regression	$Y \;=\; f(X, \beta) + \;\varepsilon$	Equation 7

The MLP algorithm operates as follows:

1. Similar to the perceptron, the MLP processes input data and parameters between the input and hidden layer which undergo partial derivatives, resulting in a value in the hidden layer that is not incremented, unlike the behaviour of an activation function
2. Activation functions like sigmoid, rectified linear units, and tanh are utilised in the hidden layers of MLP to transfer the computed output to the visible layer."

3. After the activation function generates the anticipated output in the visible layer, the corresponding partial derivatives are extracted and transmitted to another layer within MLP.
4. Steps two and three are then iteratively repeated until the final output is achieved through this process
5. The obtained estimates serve as the output to generate results for either a feed-forward technique utilising the chosen activation methods for MLP (when working with training data), or a selection based on the results (when working with testing data)."

During training, MLP predicts labels for historical data and attempts to fit predictions to these labels to predict values for new data. The outcome of the MLP confusion matrix is presented in Figure 8.

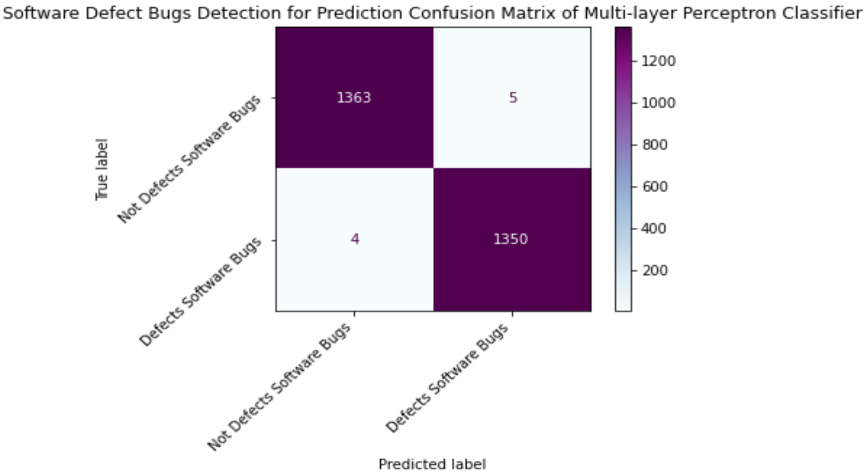


Figure 8: Confusion Matrix Multi-Layer Perceptron's (MLP) Algorithm

At the time of conducting this research, the confusion matrix was described as [[A B] [C D]], where

- A show the count of accurately predicted negative instances
- B shows the count of positive instances that were incorrectly predicted,
- C represents the number of instances that were incorrectly predicted as negative, and
- D represents the number of instances that were correctly predicted as positive.

If we assume that Perceptron's Multilayer (MLP) model is appropriate for this scenario, then the confusion matrix was useful in determining the predicted labels for our detection and prediction.

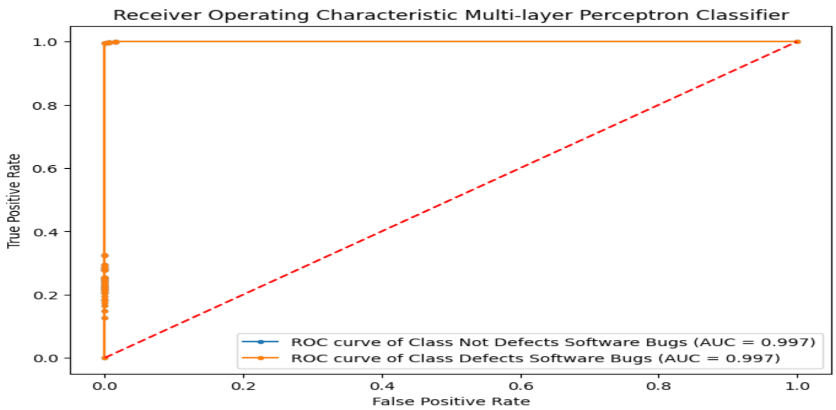


Figure 9: Receiver Operating Characteristic (ROC) Curve for Multi-Layer Perceptron's (MLP)

The results of using the Multi-Layer Perceptron's (MLP) algorithm on a synthetic dataset can be visualised through the Receiver Operating Characteristic (ROC) Curve, as shown in Figure 9. In this study, we utilised the concept of ROC curves to evaluate the accuracy of our model's predictions for user reviews ratings. This analysis allows us to better understand prediction patterns and improve the overall precision of our estimation method.

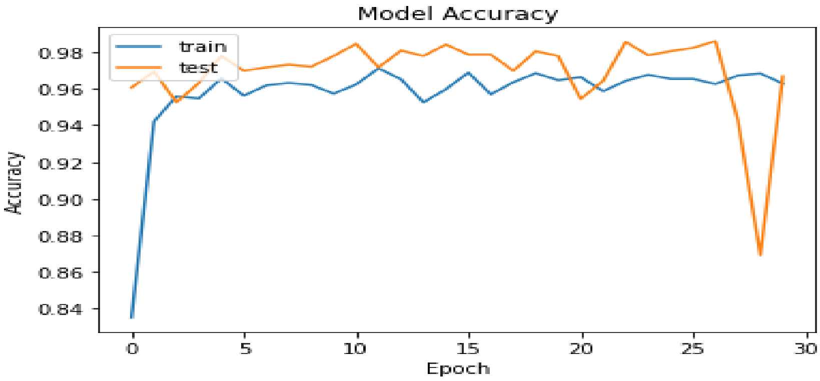


Figure 10: Model Accuracy Multi-Layer Perceptron's (MLP) Algorithm

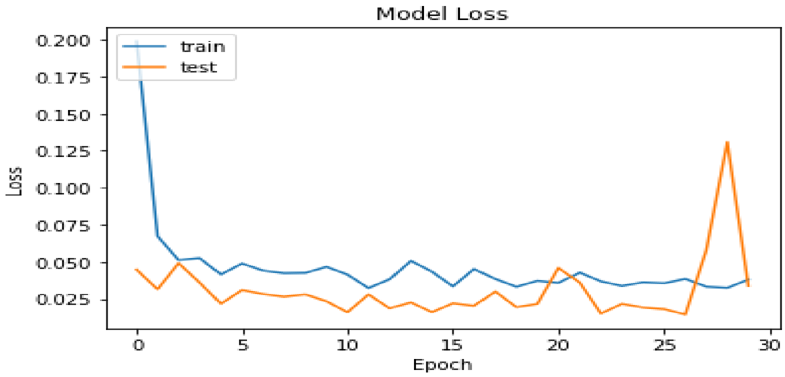


Figure 11: Model Loss Multi-Layer Perceptron's (MLP) Algorithm

To evaluate our model's performance in predicting software defects, we utilised the Multi-Layer Perceptron's (MLP) Algorithm and assessed its accuracy and loss metrics. By doing so, we aimed to improve the accuracy of our prediction approach while ensuring that it fulfils software defect prediction patterns consistently. Figures 9 and 10 depict the model accuracy and loss, respectively, which were significant indicators in our analysis. Specifically, the MLP model achieved a train accuracy of 0.97 and a test accuracy of 0.97 (Figure 9), while the train loss was 0.040 and the test loss was 0.40 (Figure 10).

5.2.2.Support Vector Machine Linear (SVML) Algorithm

The Support Vector Machine Linear (SVML) is a supervised learning approach used for regression and classification tasks. This algorithm works by partitioning mixed classes on a graph into separate groups, known as Maximum Margin Higher dimensional space. The SVML model identifies the smallest piece of data between two categories and employs various mathematical techniques such as linear, nonlinear, and kernel functions (polynomial, radial base function (RBF), and sigmoid) to achieve this separation. In particular, decision boundary support vectors are used to separate data points for different classes, with the two closest points referred to as the support vector [155-156]. The SVM technique utilises mathematical classification and regression functions such as Linear SVM, Non-linear SVM, and Kernel function.

Table 5 SVM Mathematical Equations

Linear SVM Model	***xi.xj***
SVM Non-Linear	***ϕ (xi).ϕ (xj)***

Function of Kernel	*k(xi.xj)*

The SVML algorithm follows a set of crucial steps.

1. Firstly, it identifies the appropriate hyperplanes that can effectively separate the data and maximise the margins between the different classes.
2. Additionally, it can also handle non-linearly separable data using various techniques to prevent misinterpretation.
3. Secondly, it transforms the input data into a higher dimensional space where it becomes easier to identify surface areas and make immediate selections. Finally, it restructures the challenge so that the data can be accurately transcribed to this high-dimensional space.

Once the algorithm is trained, it can be used to predict the labels for both old and new data values. The goal is to make these predictions match the actual labels as closely as possible. Figure 12 shows the resulting confusion matrix for the SVML predictions.

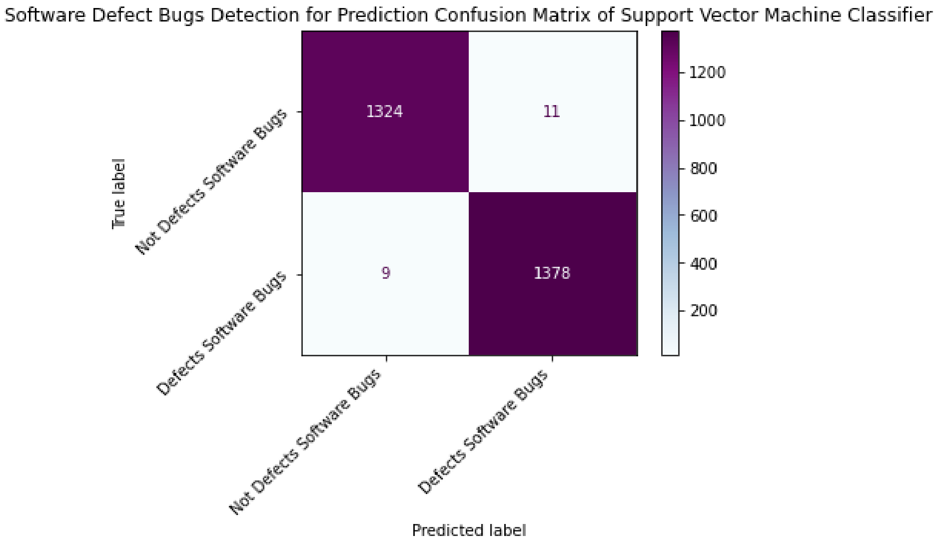


Figure 12: Confusion Matrix Support Vector Machine Linear (SVML) Algorithm

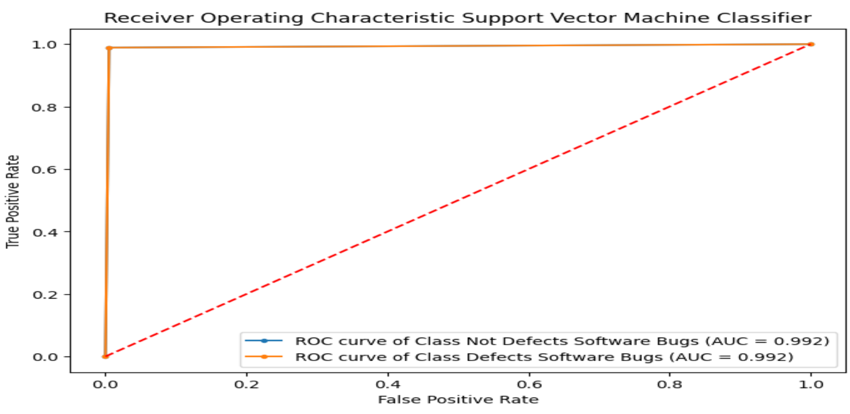


Figure 13: Confusion Matrix Support Vector Machine Linear (SVML) Algorithm

ROC analysis is a method used to evaluate how well a classifier model performs when the threshold for classifying data is changed. This analysis is closely related to cost/benefit research, where the costs and benefits of decisions are taken into consideration. Figure 13 shows the SVML ROC curve, which illustrates the performance of a support vector machine with a linear kernel at different threshold values.

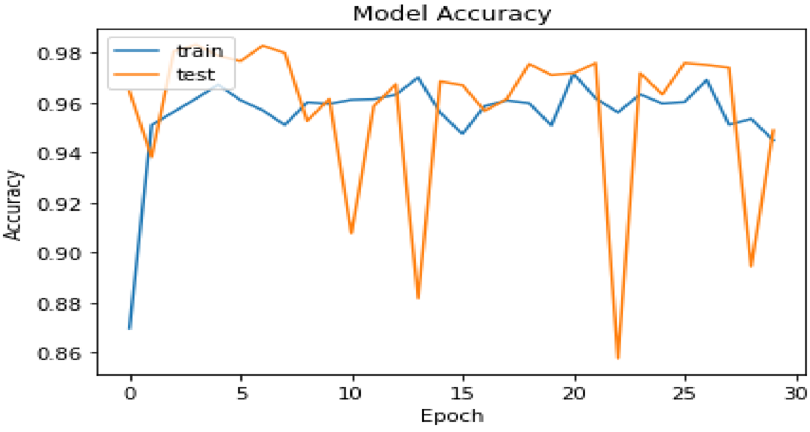


Figure 14: Model Accuracy Support Vector Machine Linear (SVML) Algorithm

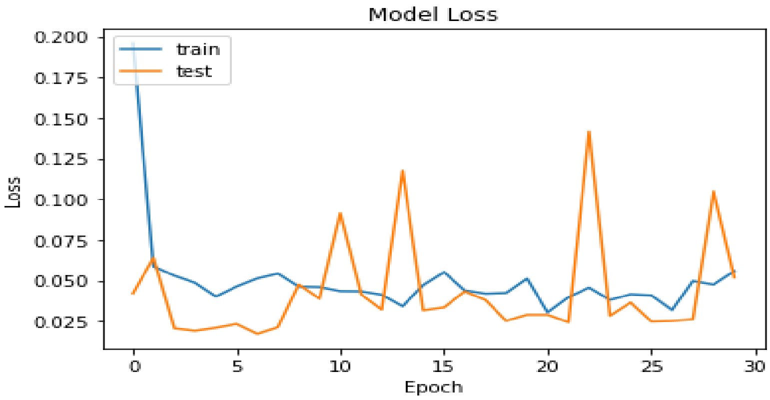


Figure 15: Model Loss Support Vector Machine Linear (SVML) Algorithm

The statement describes the performance of the Support Vector Machine with Linear (SVML) algorithm on a dataset, as shown in Figures 14 and 15. According to the statement, in Figure 14, the accuracy of the SVML algorithm was 0.96 on the training data and 0.96 on the testing data. This means that the algorithm was able to accurately classify 96% of the data points in both the training and testing sets. In Figure 15, the model loss for the SVML algorithm was 0.050 on the training data and 0.050 on the testing data. Model loss is a measure of how well the algorithm is able to predict the correct class for each input, so a lower model loss indicates better performance. Therefore, the statement suggests that the SVML algorithm performed well on both accuracy and model loss measures for this dataset.

5.2.3. Random Forest (Rf) Algorithm

The Random Forest (RF) technique is a type of machine learning method that helps to address classifier problems. It involves using various classifiers to create a complex problem-solving system that employs classifying approaches. By combining multiple categories, RF can tackle complicated issues and improve the system's efficiency. RF is based on predictions from classification trees and determines their effectiveness by making assumptions and estimating the culmination of multiple trees. As the number of nodes increases, the output improves, reducing the limitations of a Decision Tree (DT) [157-158].

1. The RF process starts by randomly selecting observations based on available data.
2. The program then creates a tree structure for each instance, and the outcomes for every tree structure are generated.
3. During this stage, each result is decided.
4. Ultimately, the prediction outcome with the highest probability is selected as the preferred result.

The RF Algorithm also employs various mathematical functions or formulas, such as Gini (Coefficient, Index, or Ratio), Entropy and Mean Squared Error (MSE) [159]. These procedures can be used as examples to evaluate the approach.

Table 6 Random Forest Mathematical Equations

Mean Squared Error (MSE)	$\frac{1}{N} = \sqrt{\sum_{i=1}^{N} \quad (xi - yi)^2}$
Gini Coefficient	$Gini = 1 \sum_{i=1}^{C} \quad (p_i)^2$
Entropy	$Entropy = \sum_{i=1}^{c} \quad - p_i * \ log_2(p_i)$

We applied the RF technique to our dataset and assigned labels to the previous data values. This helped us predict the value of the data. When we utilise the RF approach to ensure that our predictions align with the categories during preparation, the results are shown in the matrix in Figure 16.

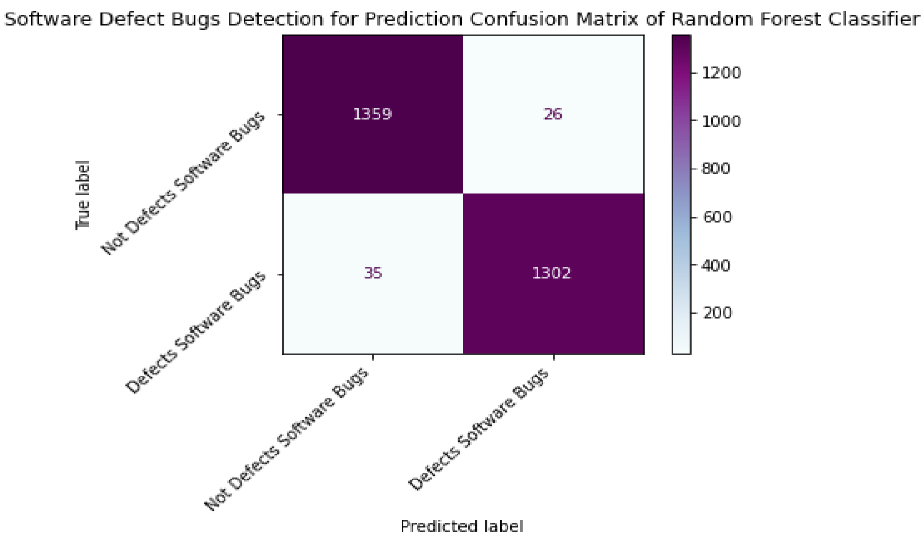


Figure 16: Confusion Matrix Random Forest (RF) Algorithm

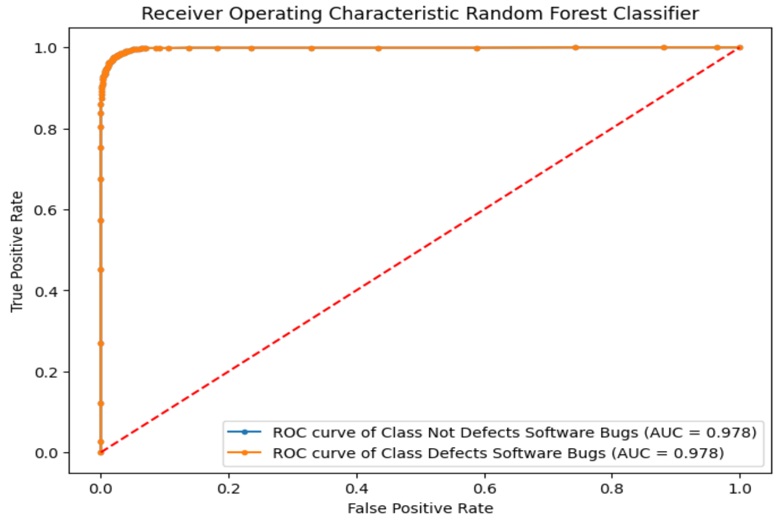
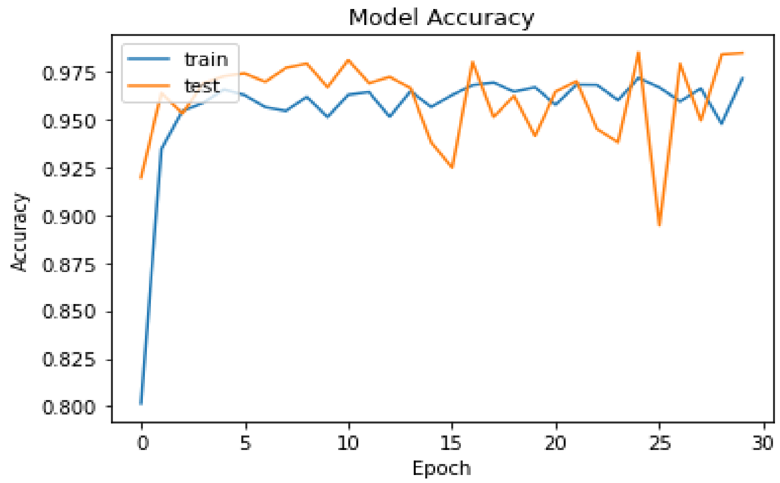


Figure 17: Random Forest (RF)Receiver Operating Characteristic (ROC Curve)

The ROC analysis is a method to evaluate the systematic performance of a classifier model when its discriminatory threshold is altered. This analysis is closely linked to cost-benefit research in making rational decisions. Figure 17 shows the result of the curve.



.

Figure 18: Model Accuracy Random Forest (RF) Algorithm

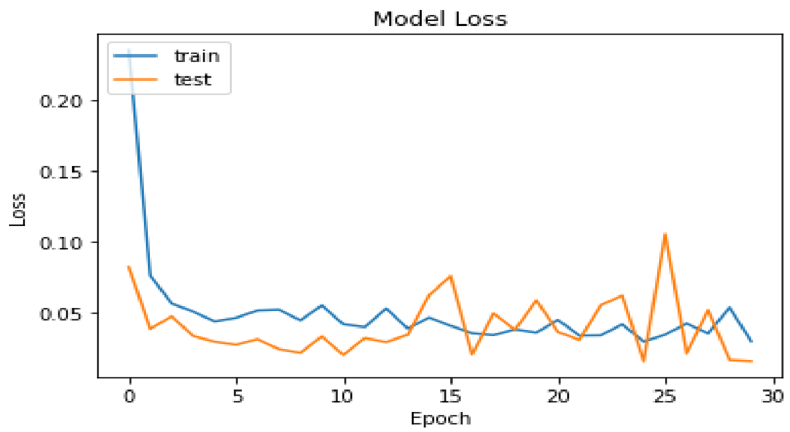


Figure 19: Model Loss Random Forest (RF) Algorithm

Figure 18, which depicts the Model Accuracy resulting from the Random Forest (RF) Algorithm, shows that the accuracy for the training data was 0.975 and for the test data was 0.976. Figure 19, which shows the Model Loss resulting from the Random Forest (RF) Algorithm, indicates that the loss for the training data was 0.04, and for the test data, it was 0.03.

6. RESULTS AND DISCUSSION

Machine learning is a practical technique that enables algorithms to tackle challenges without being explicitly programmed. Deep learning is currently the most successful form of machine learning, due to its improved processes, computing power, and access to large datasets. However, traditional machine learning techniques still play a critical role in industry applications. This study proposes an approach for predicting and detecting software defects that combines both machine learning and deep learning techniques, using data from previous software defect incidents. Our research examines the characteristics of individuals who have experienced software defects and the types of defects that they are likely to encounter. To identify software defects accurately, we combine multiple algorithms, including K-Means, Multi-layer Perceptron (MPL), K-Means, Support Vector Machines Linear (SVML), and K-Means, Random Forest (RF). Our most accurate combination of methods is achieved by combining K-Means and Multi-layer Perceptron (MPL), followed by K-Means and Support Vector Machines Linear (SVML), and K-Means and Random Forest (RF) as the third-ranked combination. The accuracy and other performance parameters of each combination are presented in Table 6 and Table 7.

Table 7 Accuracy of Models that use a Combination of Algorithms for Predicting Software Defects

Hybrid Algorithm	**Accuracy of Algorithms**
Mini-Batch K-means [156]	63.57%
Perceptron [156]	71.87%
PAC [160]	77.53%
GNB [156]	81.50%
KNN [156]	82.82%
QDA [156]	83.02%
GMM [156]	83.26%
LGBM [156]	85.99%
ET [156]	87.76%
XGBoost[156]	88.14%
RF [156]	88.18%
MVC [156]	88.27%
STC [156]	88.63
K-Means, Random Forest (RF) **Proposed Method**	97.7590007347538
K-Means, Support Vector Machine (SVM) **Proposed Method**	99.1917707567964
K-Means, Multi-layer Perceptron (MLP)	99.669360764144

Proposed Method	

Table 8 Combination of Algorithms Parameter Score for Software Defects Prediction (SDP)

S/No.	Parameter Score	K-Means, RF Algorithm	K-Means, SVM Algorithm	K-Means, MLP Algorithm
1	Precision	0.97765704	0.99193050	0.99669192
2	Recall	0.97752471	0.99192200	0.99669540
3	F1-Score	0.97758017	0.99191769	0.99669353
4	Sensitivity	0.98122743	0.99484915	0.99634502
5	Specificity	0.97382198	0.98899486	0.99704579

According to the findings depicted in Figure 20 and Figure 21, it is evident that the K-Means and Multi-layer Perceptron (MLP) combination has achieved the highest accuracy level possible. Nevertheless, the combination of K-Means and Support Vector Machines Linear (SVML) ranked second, with the combination of K-Means and Random Forest (RF) ranking third

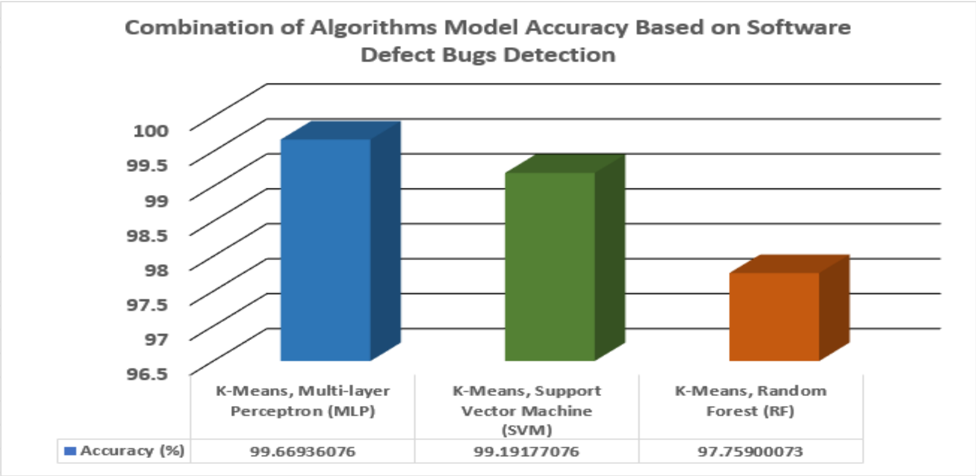


Figure 20: Combination of Algorithms Model Accuracy Software Defects Prediction (SDP) of Prediction

Based on the results, the graph indicating accuracy levels also shows that the predictions made by the combinations are at their maximum

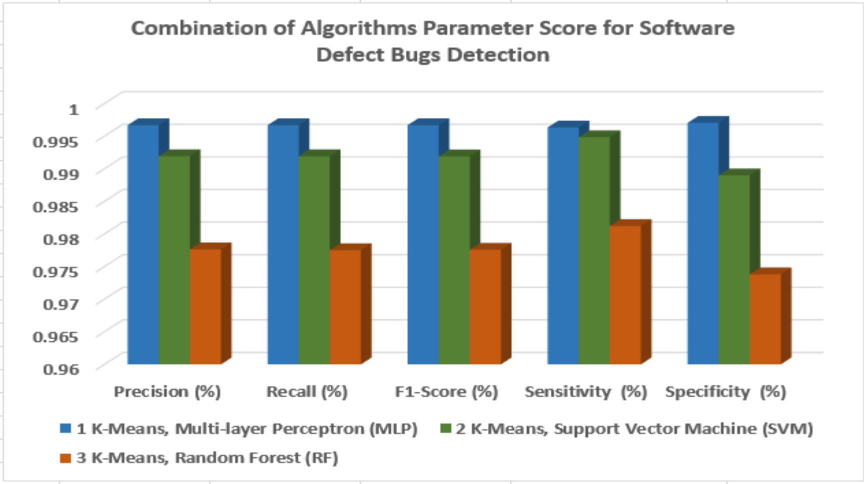


Figure 21: Combination of Algorithms Parameter Score Software Defects Prediction (SDP)

We have the ability to adjust or restrict the level of accuracy depending on our needs. For example, the Parameter Score Precision, Recall, F1-Score Sensitivity, and Specificity are currently achieving optimal accuracy.

7. CONCLUSION

This study explored the Software Defects Prediction (SDP) models by using Mathematical Modelling & Simulation methods. Many organizations use defects predicting software's in their critical operations like aviation's, healthcare services, manufacturing operations and robotics. Sometimes, it is very difficult for these organizations to predict the defect accurately before software deployment and therefore this is the matter of great concern for them.It is concluded that SDP will remain the good area for research because despite many studies during the past three decades to utilise machine learning techniques, none of the methods have demonstrated consistent reliability. It is also concluded SDP is attractive area of research as it focus on identifying flaws in software applications and proposing innovative methods to address them. It is also concluded that with the increasing use of software's in the routine operations of our corporate & social life, the need for maintainable, high-quality, and cost-effective software becomes increasingly vital. It is observed that early detection of defects makes good impact to facilitate prompt rectification which then lead to improved software reliability and performance. The current models of SDP rely on static program metrics for machine learning classifiers, but manual feature engineering may miss vital information impacting defect prediction accuracy. This study initially explores the past SDP results then aims to develop methods by adapting to future anomaly detection techniques. The study explores the various approaches of SDP which include K-Means methodology, Support Vector Machines Linear (SVML), Random Forest (RF) & Multi-layer Perceptron (MLP) algorithms and discussed the current models of SDP. The proposed SDP models are rigorously evaluated by using metrics like false alarm rate, precision, and detection rate. The results show high accuracy for K-Means and MLP (99.67%), K-Means and SVML (99.19%), and K-Means and RF (97.76%) for defect prediction.

Acknowledgment

The authors of the present research would like to acknowledge the services of Kaggle, which is a platform hosting various machine learning datasets. The accessed data of Kaggle was very helpful in the evaluating the performance of current models of Software defects prediction (SDP). We are very thankful to our friends, teachers, professional, colleagues and well wishers at the department, university and fields. We are also very thankful to the administrative and technical support from the department, university for their cooperation and support. We are especially thankful to the HEC Pakistan digital library services to provide the free access to the valuable databases and relevant books, magazines and journals.

Conflict of Interest

There was no conflict of interest among the authors of the present research paper.

References

[1] J. Tian and M. V Zelkowitz, “Complexity measure evaluation and selection,” *IEEE Trans. Softw. Eng.*, vol. 21, no. 8, pp. 641–650, 1995.

[2] K. S. Kavya and D. Y. Prasanth, “An ensemble deepboost classifier for software defect prediction,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 2021–2028, 2020.

[3] R. B. Jadhav, S. D. Joshi, U. G. Thorat, and A. S. Joshi, “A software defect learning and analysis utilizing regression method for quality software development,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 4, pp. 1275–1282, 2019.

[4] M. K. Albzeirat, M. I. Hussain, R. Ahmad, F. M. Al-Saraireh, and I. Ahmad, “A novel mathematical logic for improvement using lean manufacturing practices,” *J. Adv. Manuf. Syst.*, vol. 17, no. 03, pp. 391–413, 2018.

[5] A. G. Liu, E. Musial, and M.-H. Chen, “Progressive reliability forecasting of service-oriented software,” in *2011 IEEE international conference on web services*, IEEE, 2011, pp. 532–539.

[6] T. Menzies, Z. Milton, B. Turhan, B. Cukic, Y. Jiang, and A. Bener, “Defect prediction from static code features: current results, limitations, new approaches,” *Autom. Softw. Eng.*, vol. 17, pp. 375–407, 2010.

[7] E. Erturk and E. A. Sezer, “A comparison of some soft computing methods for software fault prediction,” *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1872–1879, 2015.

[8] M. K. Albzeirat, M. I. Hussain, R. Ahmad, F. M. Al-Saraireh, A. Salahuddin, and N. Bin-Abdun, “Applications of nano-fluid in nuclear power plants within a future vision,” *Int. J. Appl. Eng. Res.*, vol. 13, no. 7, pp. 5528–5533, 2018.

[9] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, “Benchmarking classification models for software defect prediction: A proposed framework and novel findings,” *IEEE Trans. Softw. Eng.*, vol. 34, no. 4, pp. 485–496, 2008.

[10] M. Singh and D. S. Salaria, “Software defect prediction tool based on neural network,” *Int. J. Comput. Appl.*, vol. 70, no. 22, 2013.

[11] X. Tan, X. Peng, S. Pan, and W. Zhao, “Assessing software quality by program clustering and defect prediction,” in *2011 18th working conference on Reverse Engineering*, IEEE, 2011, pp. 244–248.

[12] U. K. Mughal, M. A. Khan, P. Kumar, and S. Kumar, “Identification and Analysis of Stitching Defects at the Stitching Unit: A Case Study,” in *Proceedings of the First Central American and Caribbean International Conference on Industrial Engineering and Operations Management, Port-au-Prince, Haiti, June 15-16, 2021*, 2021. [Online]. Available: http://ieomsociety.org/proceedings/2021haiti/298.pdf

[13] M. A. Khan, A. Khatri, and H. B. Marri, “Identification of Defects in Various Processes of Spinning: A Case Study of Kotri, Sindh, Pakistan,” in *Proceedings of the First Central American and Caribbean International Conference on Industrial Engineering and Operations Management, Port-au-Prince, Haiti, June 15-16, 2021*, 2021. [Online]. Available: http://ieomsociety.org/proceedings/2021haiti/299.pdf

[14] P. Kumar, M. A. Khan, U. K. Mughal, and S. Kumar, “Exploring the Potential of Six Sigma (DMAIC) in Minimizing the Production Defects,” in *Proceedings of the 3rd International Conference on Industrial & Mechanical Engineering and Operations Management Dhaka, Bangladesh, December 26-27, 2020*, 2020. [Online]. Available: http://www.ieomsociety.org/imeom/260.pdf

[15] A. Memon, A. A. Siddiqui, and M. A. Khan, “Impact of Total Quality Management, Entrepreneurial Orientation and Organizational Excellence on Organizational Performance: Evidence from Manufacturing Firms of Kotri (S.I.T.E) Sindh Pakistan,” *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 4, no. 12, pp. 2083–2097, 2022, [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_12_december_2022/32250/final/fin_irjmets1676015268.pdf

[16] N. Baladi, P. B. Channar, L. A. Rahoo, T. Ahmed, and M. A. Khan, “Improve Customer Retention through Service Quality Attributes in the Restaurant Industry of Pakistan,” *J. Contemp. Issues Bus. Gov.*, vol. 27, no. 6, pp. 331–340, 2021, [Online]. Available: https://www.cibgp.com/article_12147_76fd80af7f9013320f57d25d1cfccea1.pdf

[17] L. A. Rahoo, M. A. K. Nagar, and A. Bhutto, “The Use of Information Retrieval Tools by the Postgraduate Students of Higher Educational Institutes of Pakistan,” *Asian J. Contemp. Educ.*, vol. 3, no. 1, pp. 59–64, 2019, doi: 10.18488/journal.137.2019.31.59.64.

[18] L. A. Rahoo, P. B. Channar, and M. A. Khan, “Analysis of Stress on the Employees of Software Development Industries of Pakistan,” *Int. Res. J. Comput. Sci. Technol.*, vol. 1, no. 1, pp. 6–12, 2020, [Online]. Available: http://irjcst.com/index.php/irjcst/article/view/2/1

[19] M. Memon, M. A. Khan, and L. A. Rahoo, "Usage and Availability of Information and Communication Technology Applications Facilities at Central Library," *Int. Res. J. Comput. Sci. Technol.*, vol. 1, no. 1, pp. 86–92, 2020, [Online]. Available: http://irjcst.com/index.php/irjcst/article/view/7/6

[20] L. A. Rahoo, P. Hasnain, A. M. Abbasi, T. Ahmed, and M. A. Khan, "The Relationship Between Information Technology and Organizational Culture in The University Libraries of Sindh, Pakistan," *J. Contemp. Issues Bus. Gov. Vol*, vol. 27, no. 2, 2021, [Online]. Available: https://www.cibgp.com/article_10816_ff2852c7bcdca4f3c72857a4da607bbe.pdf

[21] S. Arshad, H. A. Rehman, L. A. Rahoo, and M. A. K. Nagar, "Information Communication Technology Applications used to Enhance Knowledge Management in the University Libraries of Pakistan," in *Proceedings of IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8629133/

[22] K. Khan, M. A. Khan, J. A. Thebo, T. Ahmed, and L. A. Rahoo, "Examining The Human Resource Architecture Relationship With Employee Productivity Of Chemical Industries," *J. Contemp. Issues Bus. Gov.*, vol. 27, no. 2, pp. 5847–5856, 2021, [Online]. Available: https://www.cibgp.com/article_11267_91767391154f6eee74a8fa4a1c11a1c6.pdf

[23] S. Rajput, M. A. Khan, S. Samejo, G. Murtaza, and R. A. Ali, "Productivity Improvement by the Implementation of lean manufacturing practice (takt time) in an automobile assembling plant," in *Proceedings of the International Conference on Industrial Engineering and Operations Management Dubai, UAE, March 10-12, 2020*, 2020, pp. 1618–1619. [Online]. Available: http://www.ieomsociety.org/ieom2020/papers/190.pdf

[24] Z. Iftikhar *et al.*, "Productivity Improvement of Assembly Line in Textile Stitching Unit by Lean Techniques of Line Balancing and Time and Motion Study," *Int. J. Sci. Eng. Investig.*, vol. 11, no. 127, pp. 51–60, 2022, [Online]. Available: http://www.ijsei.com/papers/ijsei-1112722-07.pdf

[25] Z. Iftikhar, M. A. Khan, R. Kumar, K. Bux, and A. Haseeb, "Productivity Improvement of Garments Industry by Assembly Line Technique of Lean Manufacturing," in *Proceedings (Abstract) of the International Conference on Industrial & Mechanical Engineering and Operations Management Dhaka, Bangladesh, December 26-27, 2021.*, 2021, p. 908. [Online]. Available: https://ieomsociety.org/proceedings/2021dhaka/497.pdf

[26] M. Bukhsh *et al.*, "Productivity Improvement in Textile Industry using Lean Manufacturing Practice of Single Minute Die Exchange (SMED)," in *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management Singapore, March 7-11, 2021*, 2021. [Online]. Available: http://www.ieomsociety.org/singapore2021/papers/1282.pdf

[27] N. Jaleel, M. A. Khan, M. Jamal, M. Safeeruddin, M. M. Shajee, and U. Mughal, "Productivity Improvement by Lean Methodologies at Dyeing & Printing Plant," in *Proceedings (Abstract) of the International Conference on Industrial & Mechanical Engineering and Operations Management Dhaka, Bangladesh, December 26-27, 2021.*, 2021, p. 905. [Online]. Available: https://ieomsociety.org/proceedings/2021dhaka/495.pdf

[28] Z. Iftikhar *et al.*, "Lean Manufacturing Tools and Techniques for the Productivity Improvement in Assembly Lines Operations of Industries," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 4, no. 7, pp. 4554–4562, 2022, [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_7_july_2022/28986/final/fin_irjmets1663258443.pdf

[29] N. Li, M. Shepperd, and Y. Guo, "A systematic review of unsupervised learning techniques for software defect prediction," *Inf. Softw. Technol.*, vol. 122, p. 106287, 2020.

[30] M. S. Arain, M. A. Khan, and M. A. Kalwar, "Optimization of Target Calculation Method for Leather Skiving and Stamping: Case of Leather Footwear Industry," *Int. J. Bus. Educ. Manag. Stud.*, vol. 7, no. 1, pp. 15–30, 2020, [Online]. Available: https://www.ijbems.com/doc/IJBEMS-137.pdf

[31] M. A. Kalwar and M. A. Khan, "Increasing performance of footwear stitching line by installation of auto-trim stitching machines," *J. Appl. Res. Technol. Eng.*, vol. 1, no. 1, p. 31, 2020, doi: 10.4995/jarte.2020.13788.

[32] M. A. Kalwar and M. A. Khan, "Optimization of Procurement & Purchase Order Process in Foot Wear Industry by Using VBA in Ms Excel," *Int. J. Bus. Educ. Manag. Stud.*, vol. 6, no. 1, pp. 213–220, 2020, [Online]. Available: https://ijbems.com/doc/IJBEMS-124.pdf

[33] M. A. Kalwar, S. A. Shaikh, M. A. Khan, and T. S. Malik, "Optimization of Vendor Rate Analysis Report Preparation Method by Using Visual Basic for Applications in Excel (Case Study of Footwear Company of Lahore)," *Proc. Int. Conf. Ind. Eng. Oper. Manag. (IEOM, Dhaka, Bangladesh, December 26-27.*, 2020, [Online]. Available: https://ieomsociety.org/proceedings/2021dhaka/228.pdf

[34] M. A. Kalwar and M. A. Khan, “Optimization of Procurement & Purchase Order Process in Foot Wear Industry by Using VBA in Ms Excel,” *Int. J. Bus. Educ. Manag. Stud.*, vol. 5, no. 2, pp. 80–100, 2020.

[35] M. A. Kalwar, H. B. Marri, and M. A. Khan, “Performance Improvement of Sale Order Detail Preparation by Using Visual Basic for Applications: A Case Study of Footwear Industry,” *Int. J. Bus. Educ. Manag. Stud.*, vol. 3, no. 1, pp. 1–22, 2021, [Online]. Available: https://ijbems.com/doc/IJBEMS-159.pdf

[36] M. A. Khan, M. A. Kalwar, A. J. Malik, T. S. Malik, and A. K. Chaudhry, “Automation of Supplier Price Evaluation Report in MS Excel by Using Visual Basic for Applications: A Case of Footwear Industry,” *Int. J. Sci. Eng. Investig.*, vol. 10, no. 113, pp. 49–60, 2021, [Online]. Available: http://www.ijsei.com/papers/ijsei-1011321-08.pdf

[37] M. A. Khan, M. A. Kalwar, and A. K. Chaudhry, “Optimization of material delivery time analysis by using Visual Basic for applications in Excel,” *J. Appl. Res. Technol. Eng.*, vol. 2, no. 2, p. 89, 2021, doi: 10.4995/jarte.2021.14786.

[38] M. A. Kalwar, M. A. Khan, M. F. Shahzad, M. H. Wadho, and H. B. Marri, “Development of linear programming model for optimization of product mix and maximization of profit: case of leather industry,” *J. Appl. Res. Technol. Eng.*, vol. 3, no. 1, pp. 67–78, 2022, doi: 10.4995/jarte.2022.16391.

[39] M. A. Kalwar, M. F. Shahzad, M. H. Wadho, M. A. Khan, and S. A. Shaikh, “Automation of order costing analysis by using Visual Basic for applications in Microsoft Excel,” *J. Appl. Res. Technol. Eng.*, vol. 3, no. 1, pp. 29–59, 2022, doi: 10.4995/jarte.2022.16390.

[40] M. A. Kalwar, A. N. Wassan, M. A. Khan, M. H. Wadho, S. A. Shaikh, and H. B. Marri, “Automation of production plan generating workbook at leather footwear company of Lahore Pakistan by using VBA in Microsoft Excel,” *J. Appl. Res. Technol. Eng.*, vol. 4, no. 2, 2023, [Online]. Available: https://polipapers.upv.es/index.php/JARTE/article/view/18941/15876

[41] A. K. Chaudhry, M. A. Kalwar, M. A. Khan, and S. A. Shaikh, “Improving the Efficiency of Small Management Information System by Using VBA,” *Int. J. Sci. Eng. Investig.*, vol. 10, no. 111, pp. 7–13, 2021, [Online]. Available: http://www.ijsei.com/papers/ijsei-1011121-02.pdf

[42] M. A. Kalwar, A. N. Wassan, Z. Phul, and M. A. Wadho, Muzamil Hussain; Malik, Tanveer Sarwar; Khan, “Automation of material cost comparative analysis report using VBA Excel: a case of footwear company of Lahore,” *J. Appl. Res. Technol. Eng.*, vol. 4, no. 1, pp. 13–23, 2023, [Online]. Available: https://polipapers.upv.es/index.php/JARTE/article/view/18776/15616

[43] M. A. Khan, S. A. Khaskheli, H. A. Kalwar, M. A. Kalwar, H. B. Marri, and M. Nebhwani, “Improving the Performance of Reception and OPD by Using Multi-Server Queuing Model in Covid-19 Pandemic,” *Int. J. Sci. Eng. Investig.*, vol. 10, no. 113, pp. 20–29, 2021.

[44] S. A. Khaskheli, H. A. Kalwar, M. A. Kalwar, H. B. Marri, M. A. Khan, and M. Nebhwani, “Application of Multi-Server Queuing Model to Analyze The Queuing System of OPD During COVID-19 Pandemic: A Case Study,” *J. Contemp. Issues Bus. Gov.*, vol. 27, no. 05, pp. 1351–1367, 2021, doi: 10.47750/cibg.2021.27.05.094.

[45] I. E. Haines and M. P. Jones, “When a system breaks: a queuing theory model for the number of intensive care beds needed during the COVID-19 pandemic,” *Med. J. Aust*, 2020.

[46] H. D. D. Meares and M. P. Jones, “When a System Breaks: A Queuing Theory Model for the Number of Intensive Intensive Intensive Care Beds Needed Dur-ing the COVID-19 Pandemic”.

[47] H. Mittal and N. Sharma, “A probabilistic model for the assessment of queuing time of coronavirus disease (COVID-19) patients using queuing model,” *Technology*, vol. 11, no. 8, pp. 22–31, 2020.

[48] S. L. Zimmerman, A. R. Rutherford, A. van der Waall, M. Norena, and P. Dodek, “A queuing model for ventilator capacity management during the COVID-19 pandemic,” *Health Care Manag. Sci.*, pp. 1–17, 2023.

[49] M. A. Kalwar, H. B. Marri, M. A. Khan, and S. A. Khaskheli, “Applications of Queuing Theory and Discrete Event Simulation in Health Care Units of Pakistan,” *Int. J. Sci. Eng. Investig.*, vol. 10, no. 9, pp. 6–18, 2021, [Online]. Available: www.IJSEI.com

[50] S. A. Khaskheli, H. B. Marri, M. Nebhwani, M. A. Khan, and M. Ahmed, “Compartive study of queuing systems of medical out patient departments of two public hospitals,” *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, vol. 0, no. March, pp. 2702–2720, 2020.

[51] A.-T. Nguyen, S. Reiter, and P. Rigo, “A review on simulation-based optimization methods applied to building performance analysis,” *Appl. Energy*, vol. 113, pp. 1043–1058, 2014.

[52] H. Koziolek, “Performance evaluation of component-based software systems: A survey,” *Perform. Eval.*, vol. 67, no. 8, pp. 634–658, 2010.

[53] Y. Dutil, D. R. Rousse, N. Ben Salah, S. Lassue, and L. Zalewski, “A review on phase-change materials: Mathematical modeling and simulations,” *Renew. Sustain. Energy Rev.*, vol. 15, no. 1, pp. 112–130, 2011.

[54] M. Z. Khan and R. Alluhaibi, “Performance Analysis of Software Defects Prediction using Over-Sampling (SMOTE) and Resampling,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 11, pp. 202–215, 2019.

[55] W. Ahmad, A. Rasool, A. R. Javed, T. Baker, and Z. Jalil, “Cyber security in IoT-based cloud computing: A comprehensive survey,” *Electronics*, vol. 11, no. 1, p. 16, 2021.

[56] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, “Ai meta-learners and extra-trees algorithm for the detection of phishing websites,” *IEEE access*, vol. 8, pp. 142532–142542, 2020.

[57] L. Tang and Q. H. Mahmoud, “A survey of machine learning-based solutions for phishing website detection,” *Mach. Learn. Knowl. Extr.*, vol. 3, no. 3, pp. 672–694, 2021.

[58] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, “A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment,” *Comput. Commun.*, vol. 175, pp. 47–57, 2021.

[59] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, “A comprehensive survey of AI-enabled phishing attacks detection techniques,” *Telecommun. Syst.*, vol. 76, pp. 139–154, 2021.

[60] V. Gaur and R. Kumar, “Analysis of machine learning classifiers for early detection of DDoS attacks on IoT devices,” *Arab. J. Sci. Eng.*, vol. 47, no. 2, pp. 1353–1374, 2022.

[61] P. K. Sadhu, V. P. Yanambaka, and A. Abdelgawad, “Internet of things: Security and solutions survey,” *Sensors*, vol. 22, no. 19, p. 7433, 2022.

[62] M. Majid *et al.*, “Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review,” *Sensors*, vol. 22, no. 6, p. 2087, 2022.

[63] C. Gupta, I. Johri, K. Srinivasan, Y.-C. Hu, S. M. Qaisar, and K.-Y. Huang, “A systematic review on machine learning and deep learning models for electronic information security in mobile networks,” *Sensors*, vol. 22, no. 5, p. 2017, 2022.

[64] M. Shafiq and Z. Gu, “Deep residual learning for image recognition: A survey,” *Appl. Sci.*, vol. 12, no. 18, p. 8972, 2022.

[65] A. Safi and S. Singh, “A systematic literature review on phishing website detection techniques,” *J. King Saud Univ. Inf. Sci.*, 2023.

[66] S. Balsamo, A. Di Marco, P. Inverardi, and M. Simeoni, “Model-based performance prediction in software development: A survey,” *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 295–310, 2004.

[67] Y.-T. Li and S. Malik, “Performance analysis of embedded software using implicit path enumeration,” *IEEE Trans. Comput. Des. Integr. circuits Syst.*, vol. 16, no. 12, pp. 1477–1487, 1997.

[68] C.-Y. Huang, “Performance analysis of software reliability growth models with testing-effort and change-point,” *J. Syst. Softw.*, vol. 76, no. 2, pp. 181–194, 2005.

[69] R. Garg, K. Sharma, R. Kumar, and R. K. Garg, “Performance analysis of software reliability models using matrix method,” *Int. J. Comput. Inf. Eng.*, vol. 4, no. 11, pp. 1646–1653, 2010.

[70] S. K. Punia, M. Kumar, T. Stephan, G. G. Deverajan, and R. Patan, “Performance analysis of machine learning algorithms for big data classification: Ml and ai-based algorithms for big data analysis,” *Int. J. E-Health Med. Commun.*, vol. 12, no. 4, pp. 60–75, 2021.

[71] M. Nabi, A. Wahid, and P. Kumar, “Performance Analysis of Classification Algorithms in Predicting Diabetes.,” *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 3, 2017.

[72] P. Pahwa, M. Papreja, and R. Miglani, “Performance analysis of classification algorithms,” *Int J Comput Sci Mob Comput*, vol. 3, no. 4, pp. 50–58, 2014.

[73] E. v Venkatesan and T. Velmurugan, “Performance analysis of decision tree algorithms for breast cancer classification,” *Indian J. Sci. Technol.*, vol. 8, no. 29, pp. 1–8, 2015.

[74] S. Vanaja and K. Rameshkumar, “Performance analysis of classification algorithms on medical diagnoses-a survey,” *J. Comput. Sci.*, vol. 11, no. 1, p. 31, 2015.

[75] M. Abdar, M. Zomorodi-Moghadam, R. Das, and I.-H. Ting, “Performance analysis of classification algorithms on early detection

of liver disease," *Expert Syst. Appl.*, vol. 67, pp. 239–251, 2017.

[76] J. Pachouly, S. Ahirrao, K. Kotecha, G. Selvachandran, and A. Abraham, "A systematic literature review on software defect prediction using artificial intelligence: Datasets, Data Validation Methods, Approaches, and Tools," *Eng. Appl. Artif. Intell.*, vol. 111, p. 104773, 2022.

[77] R. S. Wahono, "A systematic literature review of software defect prediction," *J. Softw. Eng.*, vol. 1, no. 1, pp. 1–16, 2015.

[78] Z. Li, X.-Y. Jing, and X. Zhu, "Progress on approaches to software defect prediction," *Iet Softw.*, vol. 12, no. 3, pp. 161–175, 2018.

[79] M. K. Thota, F. H. Shajin, and P. Rajesh, "Survey on software defect prediction techniques," *Int. J. Appl. Sci. Eng.*, vol. 17, no. 4, pp. 331–344, 2020.

[80] V. U. B. Challagulla, F. B. Bastani, I.-L. Yen, and R. A. Paul, "Empirical assessment of machine learning based software defect prediction techniques," *Int. J. Artif. Intell. Tools*, vol. 17, no. 02, pp. 389–400, 2008.

[81] M. Jorayeva, A. Akbulut, C. Catal, and A. Mishra, "Machine learning-based software defect prediction for mobile applications: A systematic literature review," *Sensors*, vol. 22, no. 7, p. 2551, 2022.

[82] N. E. Fenton and M. Neil, "A critique of software defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 25, no. 5, pp. 675–689, 1999.

[83] T. Bergander, Y. Luo, and A. Ben Hamza, "Software defects prediction using operating characteristic curves," in *2007 IEEE International Conference on Information Reuse and Integration*, IEEE, 2007, pp. 713–718.

[84] K. Jeet, N. Bhatia, and R. S. Minhas, "A bayesian network based approach for software defects prediction," *ACM SIGSOFT Softw. Eng. Notes*, vol. 36, no. 4, pp. 1–5, 2011.

[85] M. Assim, Q. Obeidat, and M. Hammad, "Software defects prediction using machine learning algorithms," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, IEEE, 2020, pp. 1–6.

[86] X. Cai, S. Geng, D. Wu, and J. Chen, "Unified integration of many-objective optimization algorithm based on temporary offspring for software defects prediction," *Swarm Evol. Comput.*, vol. 63, p. 100871, 2021.

[87] A. N. Babatunde, R. O. Ogundokun, L. B. Adeoye, and S. Misra, "Software Defect Prediction Using Dagging Meta-Learner-Based Classifiers," *Mathematics*, vol. 11, no. 12, p. 2714, 2023.

[88] Q. Zhang and J. Ren, "Software-defect prediction within and across projects based on improved self-organizing data mining," *J. Supercomput.*, vol. 78, no. 5, pp. 6147–6173, 2022.

[89] X. Yu, J. Li, and F. Kang, "SSA optimized back propagation neural network model for dam displacement monitoring based on long-term temperature data," *Eur. J. Environ. Civ. Eng.*, vol. 27, no. 4, pp. 1617–1643, 2023.

[90] S. P. Chatzis and A. S. Andreou, "Maximum entropy discrimination poisson regression for software reliability modeling," *IEEE Trans. neural networks Learn. Syst.*, vol. 26, no. 11, pp. 2689–2701, 2015.

[91] M. Prashanthi and C. M. Miryala, "Defect prediction in software using spiderhunt-based deep convolutional neural network classifier," *Int. J. Netw. Virtual Organ.*, vol. 27, no. 4, pp. 337–357, 2022.

[92] H. Carreon-Ortiz, F. Valdez, and O. Castillo, "A new discrete mycorrhiza optimization nature-inspired algorithm," *Axioms*, vol. 11, no. 8, p. 391, 2022.

[93] F. Hassan, N. A. Qureshi, M. A. Khan, Muhammad Zohaib Khan, A. S. Soomro, A. Imroz, and H. B. Marri, "An Integrated Approach for Sentiment Classification and Information Retrieval Techniques Using K-Means, Logistic Regression, Random Forest, and Decision Tree, Algorithm," *J. Appl. Res. Technol. Eng.*, vol. 4, no. 2, 2023, [Online]. Available: https://polipapers.upv.es/index.php/JARTE/article/view/19306/15859

[94] Y. Peng, G. Kou, G. Wang, W. Wu, and Y. Shi, "Ensemble of software defect predictors: an AHP-based evaluation method," *Int. J. Inf. Technol. Decis. Mak.*, vol. 10, no. 01, pp. 187–206, 2011.

[95] Y. Jiang, M. Li, and Z.-H. Zhou, "Software defect detection with ROCUS," *J. Comput. Sci. Technol.*, vol. 26, no. 2, pp. 328–342, 2011.

[96] D. Ryu, J.-I. Jang, and J. Baik, "A transfer cost-sensitive boosting approach for cross-project defect prediction," *Softw. Qual. J.*, vol. 25, pp. 235–272, 2017.

[97] S. Kabir and Y. Papadopoulos, “Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments: A review,” *Saf. Sci.*, vol. 115, pp. 154–175, 2019.

[98] M. Z. Khan *et al.*, “The Performance Analysis of Machine Learning Algorithms for Credit Card Fraud Detection,” *Int. J. Online Biomed. Eng.*, vol. 19, no. 03, pp. 82–98, 2023, doi: 10.3991/ijoe.v19i03.35331.

[99] B. A. Akinnuwesi, G. D. Adenaike, and O. C. Nwokoro, “A Systematic Review of Soft Computing Techniques for Software Testing.,” *Int. J. Comput. Sci. Manag. Stud.*, vol. 40, no. 4, 2019.

[100] P. D. Singh and A. Chug, “Software defect prediction analysis using machine learning algorithms,” in *2017 7th international conference on cloud computing, data science & engineering-confluence*, IEEE, 2017, pp. 775–781.

[101] J. Ren, K. Qin, Y. Ma, and G. Luo, “On software defect prediction using machine learning,” *J. Appl. Math.*, vol. 2014, 2014.

[102] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, “Comments on ‘researcher bias: the use of machine learning in software defect prediction,’” *IEEE Trans. Softw. Eng.*, vol. 42, no. 11, pp. 1092–1094, 2016.

[103] C. L. Prabha and N. Shivakumar, “Software defect prediction using machine learning techniques,” in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, IEEE, 2020, pp. 728–733.

[104] S. Stradowski and L. Madeyski, “Industrial applications of software defect prediction using machine learning: A business-driven systematic literature review,” *Inf. Softw. Technol.*, p. 107192, 2023.

[105] A. Khalid, G. Badshah, N. Ayub, M. Shiraz, and M. Ghouse, “Software Defect Prediction Analysis Using Machine Learning Techniques,” *Sustainability*, vol. 15, no. 6, p. 5517, 2023.

[106] I. Mehmood *et al.*, “A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning,” *IEEE Access*, 2023.

[107] X. Peng, “Research on software defect prediction and analysis based on machine learning,” in *Journal of Physics: Conference Series*, IOP Publishing, 2022, p. 12043.

[108] Z. Xu *et al.*, “LDFR: Learning deep feature representation for software defect prediction,” *J. Syst. Softw.*, vol. 158, p. 110402, 2019.

[109] S. Wang, T. Liu, J. Nam, and L. Tan, “Deep semantic feature learning for software defect prediction,” *IEEE Trans. Softw. Eng.*, vol. 46, no. 12, pp. 1267–1293, 2018.

[110] L. Qiao, X. Li, Q. Umer, and P. Guo, “Deep learning based software defect prediction,” *Neurocomputing*, vol. 385, pp. 100–110, 2020.

[111] Z. M. Zain, S. Sakri, and N. H. A. Ismail, “Application of Deep Learning in Software Defect Prediction: Systematic Literature Review and Meta-analysis,” *Inf. Softw. Technol.*, p. 107175, 2023.

[112] M. Anbu, “Improved mayfly optimization deep stacked sparse auto encoder feature selection scorched gradient descent driven dropout XLM learning framework for software defect prediction,” *Concurr. Comput. Pract. Exp.*, vol. 34, no. 25, p. e7240, 2022.

[113] M. Nevendra and P. Singh, “A Survey of Software Defect Prediction Based on Deep Learning,” *Arch. Comput. Methods Eng.*, vol. 29, no. 7, pp. 5723–5748, 2022.

[114] A. Abdu, Z. Zhai, R. Algabri, H. A. Abdo, K. Hamad, and M. A. Al-antari, “Deep learning-based software defect prediction via semantic key features of source code—systematic survey,” *Mathematics*, vol. 10, no. 17, p. 3120, 2022.

[115] F. U. Zaman, M. A. Khuhro, K. Kumar, N. Mirbahar, Z. Khan, and A. Kalhoro, “Comparative Case Study Difference Between Azure Cloud SQL and Mongo Atlas MongoDB NoSQL Database,” *Int. J. Emerg. Trends Eng. Res.*, vol. 9, no. 7, pp. 999–1002, 2021, doi: 10.30534/ijeter/2021/26972021.

[116] C. Shyamala and S. A. Sahaaya Arul Mary, “Defect prediction in medical software using hybrid genetic optimized support vector machines,” *J. Med. Imaging Heal. Informatics*, vol. 6, no. 7, pp. 1600–1604, 2016.

[117] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, “Using the support vector machine as a classification method for software defect prediction with static code metrics,” in *Engineering Applications of Neural Networks: 11th International Conference, EANN 2009, London, UK, August 27-29, 2009. Proceedings 11*, Springer, 2009, pp. 223–234.

[118] H. Can, X. Jianchun, Z. Ruide, L. Juelong, Y. Qiliang, and X. Liqiang, “A new model for software defect prediction using particle swarm optimization and support vector machine,” in *2013 25th Chinese Control and Decision Conference (CCDC)*, IEEE, 2013,

pp. 4106–4110.

[119] D. Ryu, O. Choi, and J. Baik, “Value-cognitive boosting with a support vector machine for cross-project defect prediction,” *Empir. Softw. Eng.*, vol. 21, pp. 43–71, 2016.

[120] S. Goyal, “Effective software defect prediction using support vector machines (SVMs),” *Int. J. Syst. Assur. Eng. Manag.*, vol. 13, no. 2, pp. 681–696, 2022.

[121] J. Liu, J. Lei, Z. Liao, and J. He, “Software defect prediction model based on improved twin support vector machines,” *Soft Comput.*, pp. 1–10, 2023.

[122] Q. Wang, S. Wu, and M.-S. Li, “Software defect prediction,” *J. Softw.*, vol. 19, no. 7, pp. 1565–1580, 2008.

[123] L. Gong, S. Jiang, and L. Jiang, “Tackling class imbalance problem in software defect prediction through cluster-based over-sampling with filtering,” *IEEE Access*, vol. 7, pp. 145725–145737, 2019.

[124] R. Annisa, D. Rosiyadi, and D. Riana, “Improved point center algorithm for k-means clustering to increase software defect prediction,” *Int. J. Adv. Intell. Informatics*, vol. 6, no. 3, pp. 328–339, 2020.

[125] Z. Hu and Y. Zhu, “Cross-project defect prediction method based on genetic algorithm feature selection,” *Eng. Reports*, p. e12670, 2023.

[126] A. Shankar Mishra and S. Singh Rathore, “Implicit and explicit mixture of experts models for software defect prediction,” *Softw. Qual. J.*, pp. 1–38, 2023.

[127] S. Zhang, S. Jiang, and Y. Yan, “A Software Defect Prediction Approach Based on Hybrid Feature Dimensionality Reduction,” *Sci. Program.*, vol. 2023, 2023.

[128] V. A. Phan, “Learning Stretch-Shrink Latent Representations With Autoencoder and K-Means for Software Defect Prediction,” *IEEE Access*, vol. 10, pp. 117827–117835, 2022.

[129] S. G. Jacob, “Improved random forest algorithm for software defect prediction through data mining techniques,” *Int. J. Comput. Appl.*, vol. 117, no. 23, 2015.

[130] F. Matloob *et al.*, “Software defect prediction using ensemble learning: A systematic literature review,” *IEEE Access*, vol. 9, pp. 98754–98771, 2021.

[131] W.-D. Zhao, S.-D. Zhang, and M. Wang, “Software Defect Prediction Method Based on Cost-Sensitive Random Forest,” in *International Conference on Intelligent Information Processing*, Springer, 2022, pp. 369–381.

[132] F. H. Alshammari, “Software Defect Prediction and Analysis Using Enhanced Random Forest (extRF) Technique: A Business Process Management and Improvement Concept in IOT-Based Application Processing Environment.,” *Mob. Inf. Syst.*, 2022.

[133] M. J. Hernández-Molinos, A. J. Sánchez-García, R. E. Barrientos-Martínez, J. C. Pérez-Arriaga, and J. O. Ocharán-Hernández, “Software Defect Prediction with Bayesian Approaches,” *Mathematics*, vol. 11, no. 11, p. 2524, 2023.

[134] T. Sharma, A. Jatain, S. Bhaskar, and K. Pabreja, “Ensemble Machine Learning Paradigms in Software Defect Prediction,” *Procedia Comput. Sci.*, vol. 218, pp. 199–209, 2023.

[135] M. Z. Khan, F. U. Zaman, M. Adnan, A. Imroz, and M. A. Rauf, “Comparative Case Study : An Evaluation of Performance Computation Between SQL And NoSQL Database,” *Sindh J. Headways Softw. Eng.*, vol. 01, no. 02, pp. 14–23, 2022.

[136] Y. Zhang, D. Lo, X. Xia, and J. Sun, “An empirical study of classifier combination for cross-project defect prediction,” in *2015 IEEE 39th Annual computer software and applications conference*, IEEE, 2015, pp. 264–269.

[137] I. Arora, V. Tetarwal, and A. Saha, “Open issues in software defect prediction,” *Procedia Comput. Sci.*, vol. 46, pp. 906–912, 2015.

[138] A. Iqbal, S. Aftab, and F. Matloob, “Performance analysis of resampling techniques on class imbalance issue in software defect prediction,” *Int. J. Inf. Technol. Comput. Sci*, vol. 11, no. 11, pp. 44–53, 2019.

[139] A. Iqbal and S. Aftab, “A Classification Framework for Software Defect Prediction Using Multi-filter Feature Selection Technique and MLP.,” *Int. J. Mod. Educ. Comput. Sci.*, vol. 12, no. 1, 2020.

[140] J. M. Catherine and S. Djodilatchoumy, “Multi-layer perceptron neural network with feature selection for software defect prediction,” in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, IEEE, 2021, pp. 228–232.

[141] L. Chen, C. Wang, and S. Song, “Software defect prediction based on nested-stacking and heterogeneous feature selection,” *Complex Intell. Syst.*, vol. 8, no. 4, pp. 3333–3348, 2022.

[142] M. Z. Khan *et al.*, “Comparative case Study : An Evaluation of Performance Computation between Support Vector Machine , K-Nearest Comparative Study : Evaluation of Performance Computation Between Support Vector Component Analysis,” *J. Tianjin Univ. Sci. Technol.*, no. April, 2022, doi: 10.17605/OSF.IO/HK3SF.

[143] Ş. Ay, E. Ekinci, and Z. Garip, “A comparative analysis of meta-heuristic optimization algorithms for feature selection on ML-based classification of heart-related diseases,” *J. Supercomput.*, pp. 1–30, 2023.

[144] R. Kaur, “A comparative analysis of selected set of natural language processing (NLP) and machine learning (ML) algorithms for clinical coding using clinical classification standards.” Western Sydney University (Australia), 2018.

[145] B. F. de Souza, A. C. de Carvalho, and C. Soares, “A comprehensive comparison of ml algorithms for gene expression data classification,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2010, pp. 1–8.

[146] G. Tanriver, M. Soluk Tekkesin, and O. Ergen, “Automated detection and classification of oral lesions using deep learning to detect oral potentially malignant disorders,” *Cancers (Basel).*, vol. 13, no. 11, p. 2766, 2021.

[147] R. A. Welikala *et al.*, “Automated detection and classification of oral lesions using deep learning for early detection of oral cancer,” *IEEE Access*, vol. 8, pp. 132677–132693, 2020.

[148] Datavedas, “Classification Problems,” *Datavedas Classification Problems*, 2018. https://www.datavedas.com/wp-content/uploads/2018/05/3.1.1.2-CLASSIFICATION-PROBLEMS-1.png

[149] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, “A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering?,” *Intell. Decis. Technol.*, vol. 12, no. 1, pp. 3–14, 2018.

[150] M. A. I. Aquil and W. H. W. Ishak, “Predicting software defects using machine learning techniques,” *Int. J.*, vol. 9, no. 4, pp. 6609–6616, 2020.

[151] Mustafa Cevik, “Software Defect Prediction Data Analysis,” *Kaggle*, 2019. https://www.kaggle.com/code/semustafacevik/software-defect-prediction-data-analysis/data

[152] I. Dabbura, “K-means clustering: Algorithm, applications, evaluation methods, and drawbacks,” *Towar. Data Sci.*, 2018.

[153] DeepAI, “Multilayer Perceptron,” *Mach. Learn. Gloss. Terms, Deep.*, 2020, [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron

[154] C. V. Nicholson, “A Beginner’s Guide to Multilayer Perceptrons (MLP),” *Pathmind*, 2020. https://wiki.pathmind.com/multilayer-perceptron

[155] A. A. Khan, A. A. Laghari, S. Awan, and A. K. Jumani, “Fourth industrial revolution application: network forensics cloud security issues,” *Secur. Issues Priv. Concerns Ind. 4.0 Appl.*, pp. 15–33, 2021.

[156] R. A. Laghari, J. Li, A. A. Laghari, and S. Wang, “A review on application of soft computing techniques in machining of particle reinforcement metal matrix composites,” *Arch. Comput. Methods Eng.*, vol. 27, pp. 1363–1377, 2020.

[157] Tutorialspoint, “Classification Algorithms - Random Forest,” *Machine Learning with Python, Tutorialspoint*, 2023. Classification Algorithms - Random Forest

[158] N. Mbaabu, “Introduction to Random Forest in Machine Learning,” *Section*, 2020. https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

[159] M. Schott, “Random forest algorithm for machine learning,” *Medium*, 2019.

[160] S. Karim, H. L. H. S. Warnars, F. L. Gaol, E. Abdurachman, and B. Soewito, “Software metrics for fault prediction using machine learning approaches: A literature review with PROMISE repository dataset,” in *2017 IEEE international conference on cybernetics and computational intelligence (CyberneticsCom)*, IEEE, 2017, pp. 19–23.

Authors' Profiles

Shadab Yameen Shaikh was born in Pakistan. She has completed her graduation from the Institute of Mathematics and Computer Science, University of Sindh, Jamshoro, Sindh, Pakistan. She has attended various national & international conferences. She has also participated in many professional seminars, workshops, symposia and trainings. Her research interests include Mathematical Modeling, Statistical Analysis, Simulation, Data Science and Artificial Intelligence.

Naseem Afzal Qureshi was born in Pakistan. She has completed her graduation from the Department of Computer Science, Faculty of Science, University of Karachi, Karachi, Sindh, Pakistan. She has attended various national & international conferences. She has also participated in many professional seminars, workshops, symposia and trainings. Her research interests include Data Science, Artificial Intelligence, Machine Learning, Deep Learning, Cyber Security, Internet of Things and Cloud Computing. She has authored and presented research papers at the national & international conferences and journals.

Muhammad Zohaib Khan was born in Pakistan. He has received Master degree in Computer Science from Sindh Madressatul Islam University, Karachi, Pakistan and Bachelor degree in Computer Science from the University of Sindh, Jamshoro, Pakistan. He has worked as an IT Engineer in the Department of IT, Sindh Public Procurement Regulatory Authority from 2017 to 2019. He is currently works as Software and Data Engineer, in the Department of IT, Shaheed Mohtarma Benazir Bhutto Institute of Trauma. He has authored and presented various research papers at the national & international conferences and journals. His research interests include Data Science, Artificial Intelligence, Machine Learning, Deep Learning, and the Internet of Things.

Muhammad Ali Khan was born in Pakistan and currently works as Assistant Professor in the Department of Industrial Engineering and Management, Mehran UET, Jamshoro, Sindh, Pakistan. He is pursuing his PhD in the same department. He has completed his Bachelor of Engineering, PGD and Master of Engineering in Industrial Engineering and Management. He has also completed his MBA in Industrial Management from IoBM, Karachi, Pakistan. He has authored various research papers for conferences and journals. He has participated in many professional seminars, workshops, symposia and trainings. He does research in diversified fields of Industrial Engineering. The current projects are related to Lean manufacturing, Six Sigma, Project management, Operations management; MIS and Entrepreneurship. He has also earned various certifications in his areas of research.

Aisha Imroz was born in Pakistan. She is doing Master degree in Computer Science from the Sindh Madressatul Islam University, Karachi, Pakistan. She currently works as a Software Engineer at Avanza Solutions (Pvt.) Ltd. She has attended various national & international conferences. She has also participated in many professional seminars, workshops, symposia and trainings. Her research interests include Data Science, Artificial Intelligence, Machine Learning, Deep Learning, Cyber Security, Internet of Things, Cloud Computing, and the Medical Science.

Muhammad Ahmed Kalwar was born in Pakistan and currently works as an Assistant Manager (Production) in a footwear industry. He has completed his Bachelor & Master of Engineering in Industrial Engineering and Management from the Department of Industrial Engineering and Management of Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan. During his Master of Engineering, he has also served as Teaching Assistant in the same department. He has authored and presented various research papers at the national & international conferences and journals. His areas of interest are Operations Research, Statistical Analysis and Mathematical Modeling & Simulation.