**Project Proposal:** Deep Learning Based Model for Predictive Analysis of Faulty Software Modules

**Course:** CSCE 5218

# Members

Pooja Goyal

Roshan Sah

Sanjib Paudel

Sourabh Yadav

# General Project Idea

Development projects have been through a time-consuming and expensive assessment procedure to determine the viability of software development deliverables. As project scale and complexity grow, manually finding software problems becomes a time-consuming and costly effort. As a potential substitute to conventional aberration identification, using automated predictors to work on problematic components lets the software developer dig further into the problem. Better fault determinants may frequently locate a program that may be applied to a software consistency application in this scenario. As a result, there are several verified and created base predictors. Simple predictors, particularly fault-detection capabilities, can be used with an ensemble technique to boost efficiency even further. The fundamental goal of the study is to assess the fault diagnostic usefulness of baseline and ensemble prediction models. The suggested research, which is being used on the PROMISE directories, focuses on identifying application issues using Machine Learning and Deep Learning-based categorization approaches, then assessing empirical data to determine the best match model. The purpose of the research is to show that ensemble predictors can increase fault detection accuracy to some extent.

# Problem Statement

Predicting problematic modules prematurely in the software development process is a time-consuming aspect of quality assurance in today's world. Software fault prediction (SFP) can facilitate the efficiency of testing obligation optimization by discovering software modules that are prone to many vulnerabilities. It also assists in the more effective dissemination of quality control instruments. The efficacy of various defect prediction learning algorithms varies depending on the application, rendering them unsuitable for use with unknown software projects. The early discovery of flaws is critical since the cost of repairing these flaws increases as the software development life cycle progresses (SDLC). Screening and issue fixing are time-consuming and complex tasks in software engineering. In the last decade, there have been significant advancements in this field, intending to determine how to forecast problematic components in upcoming applications. The recovery of assessment metrics from the existing dataset is the most important component in determining how effective a defect detection approach is. There are a variety of software metrics that may be used to determine if a software instance, class, or module is problematic or not. Furthermore, several retrospective investigations have found that certain subsets of system metrics might aid classifiers in producing more efficient results. The activity of software defect prediction is essential to improve the due diligence procedure's dependability. It will aid in the construction of improvised products with little cash and in a short amount of time. Deep learning techniques are regarded to be a reliable means of diagnosing software flaws early in the SDLC by discovering a hidden pattern in the historical system.

Steered deep learning methodologies involve pre-records for learning (training data). During the learning phase, these approaches develop recommendations for identifying secret data (test data). The benchmark datasets are used in the study to allow other scholars to reconcile their computational findings with their proposed approach, allowing them to assert a higher degree of certainty from their investigation that can be validated by the whole scientific community. It also enables software specialists to optimize testing efforts based on the number of defects and to find most faults early and efficiently.

# Dataset

In this intended research, the research project uses 14 software fault datasets from a variety of open-source software development projects that are available in the PROMISE data repository. For calculations, five software fault datasets from the PROMISE data collection are extracted. The use of standardized and consistent datasets, as well as the ability to replicate studies, are two main advantages of including fault datasets in research, which are collected using fully accessible software modules.

## KC

KC dataset contains the values of C++ project which are retrieved from Halstead and McCabe retrievers.

## PC

PC dataset contains the C functions. Mainly the dataset has datapoints related to flight software for earth orbiting satellite.

## AR

AR dataset contains the data points which is dependent on the Turkish Good Manufacturer company, whose core language is C.

## CM

CM1 contains the data points which are mainly related to NASA spacecraft and is written in C.

## MC

One of the NASA Metrics Data Program defect data sets. The specific type of software is unknown.

# Baselines and Evaluation Metrics

When the dataset is suitable for learning and use, it's critical to create basic baseline models that can then be improved to create hybrid and ensemble models. Naive Bayes, Logistic Regression, Random Forest, Decision Tree, Support Vector Machine, and other approaches can be constructed from the baseline standpoint. Additionally, just a few basic neural network or perceptron models may be created. Splitting the dataset into many sub-datasets and executing independent training for each dataset might further improve training views. This method may be used to create a new perceptron that combines the findings of numerous models to produce a final output.

The above-quoted picture depicts the proposed method for determining defective software modules using various code criteria and structures.

# Experiments and Results

The Anaconda framework, which employs Python as its programming language, may be used to execute the analysis. The Anaconda framework has several visualization options as well as a lot of pre-defined libraries that may help you do better analysis. For binary classification, the accuracy parameter and confusion matrix can be used to evaluate the model. To have a better understanding of prediction outcomes, we will employ Accuracy performance measurements and results such as Confusion Matrix, Precision, Recall, and F1-Score (harmonic mean of precision and recall).