Contexte  et problème traiter :

Cet article du domaine de l’intelligence artificiel, expose les analyses, expériences, et résultats d’une recherche ayant pour objectif de proposer un modèle permettant la détection de bugs dans le code source d’un logiciel le plus tôt possible dans le cycle de vie d’un projet. Et ce en se basant sur les méthodes d’apprentissage automatique pour répondre à un problème crucial dans le domaine du génie logiciel qui consiste en le développement d’un logiciel de qualité qui réponds au attentes des clients dans un délais et un budget limité.

**Les solutions existantes :**

Les méthodes actuelles d'ingénierie logicielle présentent stratégies de "prévention des bugs" plutôt que de "détection des bugs". Les méthodes de prévention de bugs sont réalisées grâce à des processus de gestion de la propagation des erreurs. Les révisions et les inspections dans ces processus sont des moyens traditionnels de réduire activement les erreurs issues de la phase de développement. Cependant elles sont généralement assez coûteuses et entraînent une augmentation considérable des coûts d’un projet.

**Les motivations derrière la solution proposée :**

Alternativement, il est proposé dans cette recherche, la mise en œuvre des techniques d'apprentissage automatique permettant la reconnaissance d’erreurs durant le cycle de développement. Ces techniques paraissent bien adaptées à l'identification et à la prédiction des bugs dans les projets logiciels.

**Principe et grandes étapes de la solution proposée**

Dans cette recherche qui consiste à prédire l'ampleur d'un éventuel bug tel que sa gravité, sa priorité, etc. en se basant sur les métriques de code. Il a été conçu un système d’apprentissage automatique ayant comme données en entrée les attributs (code métrique) de chaque module/fonction d’un projet utilisé pour les tests. Comme ces données ont de nombreuses valeurs corrélées, une analyse en composantes principales (ACP) est effectuée pour transformer les variables corrélées en variables non corrélées. Cette élimination de la corrélation permet de réduire la dimensionnalité de ces ensembles de données.

Le système est ensuite entraîné avec cet ensemble de données optimisé en utilisant les algorithmes basés sur la régression tels que l'arbre de décision (DT), le perceptron multicouche (MLP) et les fonctions de base radiales (RBF). Les sorties du système indiquent le nombre d’erreurs total par module/fonction.

**les techniques utilisées**

Le modèle proposé dans cette article est un modèle basé sur techniques de régression pour détecter et identifier les défauts logiciels. Ainsi, deux systèmes d’apprentissage en été conçu comme suit.

* Dans le système d'apprentissage initial, l'ensemble des données (original) est utilisé entièrement tandis que dans le système d'apprentissage amélioré, un ensemble de données partiel obtenu à partir de l'ensemble l'original est utilisé.
* Dans un ensemble de données donné, il existe de nombreuses métriques de code, y compris les niveaux de priorité des erreurs (la criticité des erreurs) dans chaque module. Ce qui permet de séparer puis éliminer les erreurs de priorité mineure, pour obtenir de meilleurs résultats expérimentaux
* Les expériences pour chaque algorithme d'apprentissage automatique sont répétées 200 fois (époques) et les valeurs moyennes des résultats sont prises en compte dans le calcul du taux d'amélioration.
* Pour garantir le caractère aléatoire des données, tous les ensembles de données sont régénérés à partir de l'ensemble de données d'origine en utilisant un algorithme de brassage basé sur un déplacement aléatoire des lignes de l'ensemble de données.

**Les résultats/conclusions obtenus**

Les résultats expérimentaux en utilisant les ensembles de données complets sont susceptible d’avoir un biais dans les estimations concernant les erreurs en raison de la pondération des niveaux de criticité qui n’est pas réaliser de manière égale dans l’ensemble de données. Par contre, les résultats expérimentaux obtenu en utilisant des ensembles de données qui ne contiennent pas de défauts de priorité mineure en utilisant les arbres de décisions (DT), le perceptron multicouche (MLP) et les fonctions de base radiales (RBF) sont plus satisfaisante. En effet, elles montrent que le système proposé est capable de faire des estimations plus précises en utilisant l'ensemble de données contenant uniquement les erreurs de criticité majeure. En procédant ainsi, les performances du système l'efficacité des méthodes d'apprentissage sont avancées.