Ministry of Higher Education and
Scientific Research
University of Mosul
College of Computer Science and Mathematics
Department of Software

Designing a Tool for Code Smell Detection

A Thesis Submitted to the Council of the College of Computer Science and Mathematics University of Mosul as a Partial Fulfillment of Requirements for the Degree of Master of Science in Software

By
Noura Raad Nashaat Najeeb

Supervised by
Assist. Prof. Dr. Naktal Moiad Edan Kataa

2025 A.D. **1447 A.H.**

Abstract

The presence of code smells indicates problems in the code's structure and poor design, which reduces maintainability and development, and increases the likelihood of future failures. Therefore, early detection of smells improves software quality and reduces future problems. Recently, Machine Learning (ML) and Deep Learning (DL) techniques have offered automatic code smell detection capabilities that operate more effectively by discovering complex patterns, ensuring more accurate results, and decreasing developers reliance on manual code review.

The main aim of this thesis is to highlight one of the contributions of software engineering, detecting code smells to improve software quality, by designing and implementing a tool that detects five types of code smells such as, Long Method (LM), Feature Envy (FE), Data Class (DC), Large Class (LC), and God Class (GC). In the beginning, a comprehensive study of related work was conducted to understand the code smells and find out their specifications, including specifying the most common strategies and techniques used in this field. Secondly, this thesis specifies and applies a dataset that has been collected from Zenodo's source projects in Comma-Separated Values (CSV) format. Moreover, the Python language has been used to achieve all the detection. The tool has been adopted on a deep hybrid model consisting of a Stacked Autoencoder (SAE), Convolution Neural Network (CNN), and Long Short-Term Memory (LSTM). Therefore, the hybrid model performance has been compared with six ML techniques, such as K-Nearest Neighbor (KNN), eXtreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Decision Tree (DT), and Logistic Regression (LR), as well as two DL techniques, like CNN with LSTM. Training the hybrid model was performed with 80% of the dataset, and after that, the model was judged for accuracy with the remaining 20%. The hybrid model achieved the highest accuracy of all the individual techniques it was compared to, achieving Long Method accuracy of 99.41%, Feature Envy accuracy of 85.62%, Data Class accuracy of 99.41%, Large Class accuracy of 99.01%, and God Class accuracy of 98.81%. Last but not least, this tool can support developers in improving their work and experiments, including refactoring and enhancing the quality of the code source by providing improvement suggestions.

وزارة التعليم العالي والبحث العلمي
جامعة الموصل
كلية علوم الحاسوب والرياضيات
قسم البرمجيات

تصميم اداة لكشف رائحة الكود البرمجي

رسالة مقدمة
الى مجلس كلية علوم الحاسوب والرياضيات في جامعة الموصل
كجزء من متطلبات نيل شهادة ماجستير علوم في
البرمجيات

من قبل

نورا رعد نشأت نجيب

بإشراف

أ.م.د نكتل مؤيد عيدان كاطع

١٤٤٧ هـ ٢٠٢٥ م

المستخلص

يشير وجود روائح الكود الى وجود مشاكل في بنية الكود وسوء تصميمه، مما يقلل من قابلية الصيانة والتطوير بالاضافة الى احتمالية فشل في المستقبل. لذا فان الكشف المبكر عن الروائح تحسن من جودة البرمجيات وتقلل المشاكل المستقبلية. في الآونة الأخيرة، تُتيح تقنيات التعلم الآلي وتقنيات التعلم العميق، إمكانيات الكشف التلقائي عن روائح الكود بكفاءة أكبر من خلال اكتشاف الأنماط المُعقدة، مما يضمن نتائج أكثر دقة، ويُقلل على المطورين الاعتماد على مراجعة الكود يدويًا.

الهدف الرئيسي من هذه الرسالة هو تسليط الضوء على إحدى مساهمات هندسة البرمجيات، وهي الكشف عن روائح الكود لتحسين جودة البرمجيات، وذلك من خلال تصميم وتنفيذ أداة تكتشف خمسة أنواع من روائح الكود، وهي، الطريقة الطويلة (LM)، وحسد الميزة (FE)، وفئة البيانات (DC)، والفئة الكبيرة (LC)، والفئة الضخمة جداً (GC). في البداية، أُجريت دراسة شاملة للأعمال ذات الصلة لفهم روائح الكود ومعرفة مواصفاتها، بما في ذلك تحديد الاستراتيجيات والتقنيات الأكثر شيوعًا المستخدمة في هذا المجال. ثانيًا، تحدد هذه الدراسة وتُطبق مجموعة بيانات جُمعت من مشاريع مفتوحة المصدر من موقع Zenodo بتنسيق قيم مفصولة بفواصل (CSV). علاوة على ذلك، استُخدمت لغة بايثون لتحقيق جميع عمليات الكشف. وقد تم اعتماد الأداة على نموذج هجين عميق يتكون من مُرمِّز تلقائي مكدس (SAE)، وشبكة عصبية التفافية (CNN)، وذاكرة طويلة وقصيرة المدى (LSTM). وكذلك، تمت مقارنة أداء النموذج الهجين مع ست تقنيات للتعلم الآلي، مثل خوارزمية اقرب جار (KNN)، وتعزيز التدرج الشديد (XGBoost)، وآلة دعم المتجهات (SVM)، والانحدار التدرجي العشوائي (SGD)، وشجرة القرار (DT)، والانحدار اللوجستي (LR))، بالاضافة الى تقنيتان للتعلم العميق مثل CNN و LSTM.

تم تدريب النموذج الهجين باستخدام ٨٠٪ من مجموعة البيانات، وبعد ذلك، تم اختبار وتقييم النموذج باستخدام ٢٠٪ المتبقية. حقق النموذج الهجين أعلى دقة من بين جميع التقنيات الفردية التي تمت مقارنته بها، محققًا دقة الطريقة الطويلة بنسبة ٩٩.٤١٪، ودقة حسد الميزة بنسبة ٨٥.٦٢٪، ودقة فئة البيانات بنسبة ٩٩.٤١٪، ودقة الفئة الكبيرة بنسبة ٩٩.٠١٪، ودقة الفئة الضخمة جداً بنسبة ٩٨.٨١٪. وأخيرًا وليس آخرًا، يمكن لهذه الأداة دعم المطورين لتحسين أعمالهم وتجاربهم، بما في ذلك إعادة الهيكلة وتعزيز جودة مصدر الكود من خلال تقديم اقتراحات التحسين.