بسم الله الرحمن الرحيم

∞∞∞

تم رفع هذه الرسالة بواسطة / حسام الدين محمد مغربي

بقسم التوثيق الإلكتروني بمركز الشبكات وتكنولوجيا المعلومات دون أدنى مسئولية عن محتوى هذه الرسالة.

ملاحظات : لا يوجد

AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

Computer and Systems Engineering

Software Refactoring Using Artificial Intelligence Techniques

A Thesis submitted in partial fulfilment of the requirements of the degree of

Master of Science in Electrical Engineering

(Computer and Systems Engineering)

by

Rana Samy Menshawy Othman

Bachelor of Science in Electrical Engineering

(Computer and Systems Engineering)

Faculty of Engineering, Ain Shams University, 2017

Supervised By

Prof. Dr. Ashraf Mohamed Mohamed Salem

Professor at Computer and Systems Engineering Department

Faculty of Engineering, Ain Shams University

Prof. Dr. Ahmed Hassan Youssef

Professor at Computer and Systems Engineering Department

Faculty of Engineering, Ain Shams University

Cairo - (2022)

AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

Computer and Systems Engineering

Name: **Rana Samy Menshawy**

Thesis: **Software Refactoring using Artificial Intelligence Techniques**

Degree: **Master of Science in Electrical Engineering (Computer and Systems Engineering)**

Examiners' Committee

Name and Affiliation	Signature
Prof. Reem Mohamed Reda Bahgat Electrical Engineering, Faculty of Engineering, Cairo University, and Egypt university of Informatics	…….
Prof. Hani Mohamed Kamal Mahdi Computer and Systems Engineering, Faculty of Engineering, Ain Shams University	…….
Prof. Ashraf Mohamed Mohamed Salem Computer and Systems Engineering, Faculty of Engineering, Ain Shams University	…….
Prof. Ahmed Hassan Youssef Computer and Systems Engineering, Faculty of Engineering, Ain Shams University	…….

Date: 16 June 2022

Statement

This thesis is submitted as partial fulfilment of Master of Science in Electrical Engineering, Faculty of Engineering, Ain shams University.

The author carried out the work included in this thesis, and no part of it has been submitted for a degree or a qualification at any other scientific entity.

Rana Samy Menshawy

Date: 19 July 2022

Researcher Data

Name	: Rana Samy menshawy Othman Ali Abood
Date of birth	: 30/9/1993
Place of birth	: Cairo, Egypt
Last academic degree	: Bachelor of Science in Electrical Engineering
Field of specialization	: Computer and Systems Engineering
University issued the degree	: Ain Shams University
Date of issued degree	: 2017
Current job	: Senior Software Engineer at Valeo

Abstract

Abstract--Code smells are indicators of code anomalies in poorly implemented or designed software programs. In addition to code fault-proneness and software defects, code smells may substantially hinder code maintainability in the future. The developer's mission to read code, maintain it or add new features, which can become difficult. Automatic detection of design flaws assists developers in identifying code smells in their software programs to avoid low-quality delivery, which negatively affects software deliverable deadlines. Nevertheless, the interpretation of code smells is subjective. Artificial intelligence (AI)-based detection tools have recently attracted the attention of researchers and have succeeded in eliminating the dependency on the developers' perception and experience and the need for their prior knowledge of code smell characteristics. However, the majority of existing tools, that are based on machine learning (ML) adopted approaches, are based on structural information and disregarded the importance of capturing code semantic features that reflect the complex semantic relationships in textual code. Fewer tools exploit deep learning (DL) based on the textual features approach. This paper proposes a detection system that exploits ML and DL techniques and compares them. The paper applies six DL techniques based on code semantics features extracted from software code abstract syntax trees (ASTs), and eleven different ML techniques based on code structural features. Furthermore, we propose a complementary refactoring tool that fixes the code smells detected in the detection stage, in order to form an end-to-end refactoring solution. The results showed that DL techniques are still comparable to ML techniques, the majority of ML classifiers, especially the Random Forest algorithm, achieve exceptional performance.

Keywords: Code smell, Machine Learning, Deep Learning, Semantical feature, Structural feature, Abstract syntax tree and Refactoring.

Thesis Summary

Code smells might indicate a deeper design problem in the software code, these smells can lead to costly maintenance and quality problems in addition to code fault-proneness and bugs in the near future. There are several approaches to detect those anomalies. In addition, automatic and semi-automatic tools help the developer detect the code smells instead of the manual effort.

Code smells are subjective and can be interpreted in many different ways. The automatic detection tools succeed in eliminating the dependency on developers' perception and experience and the need for their prior knowledge of the code smells characteristics. Recently machine learning (ML) and deep learning (DL) are extensively used in code smells detection.

Refactoring is an adequate technique followed by software engineers to eliminate code smells. Software refactoring modifies the internal code structure without changing its functionality and suggests the best redesign changes to be performed. Developers who apply correct refactoring sequences to remove code smells, improve the software maintenance and development time significantly. Many tools have been created to automatically or semi-automatically detect code smells and refactor them.

In this thesis, we dive into discussing what that are code smells, and the most commonly used detection and refactoring tools utilized by the previous studies. Furthermore, a detection approach based on machine learning (ML) and deep learning (DL) approaches to detect the selected code smells is proposed. Furthermore, the Extract Method refactoring technique algorithm is discussed in order to refactor the Long Method code smell.

The purpose of this thesis is to investigate both ML and DL techniques and compare the performance of their classifiers. Eleven ML algorithms are applied to the extracted code metrics, and six DL architectures are applied to the extracted

textual features of the source code. Furthermore, the Extract Method refactoring technique is investigated and applied to our tool.

The thesis is divided into six chapters as follows:

- Chapter 1: Presents an introduction to code smells and how they greatly impact the software Business. In addition, it includes the related work and our contribution.
- Chapter 2: Presents the required background about code smells, detection approaches defined by the previous studies, and the detection and refactoring tools implemented by developers to automatically or semi-automatically detect and refactor code smells. Moreover, the challenges and limitations are discussed.
- Chapter 3: Discusses the research methodology, in addition to providing an overview of the proposed system.
- Chapter 4: Describes in detail the detection approach followed; applying Machine learning techniques based on the structural features, in addition to the Deep learning techniques based on the semantical features of the source code. Moreover, evaluating and comparing the results of all the classifiers is also presented.
- Chapter 5: Presents the used tool to refactor the Long Method code smell by the Extract Method technique.
- Chapter 6: Discloses our conclusion and future work.

Acknowledgement

I would like to thank my supervisors, Prof. Dr Ashraf Salem and Prof. Dr Ahmed Hassan for guiding me throughout this research. I am grateful for their time and support.

Finally, thanks to my family, friends and work colleagues for a lifetime of support and encouragement.

Rana Menshawy, 2022