

# Convolutional Neural Network programs meta-modeling for automatic detection of design smells

Aurel Ikama<sup>1</sup>, Foutse Khomh<sup>1</sup>, Heng Li<sup>1</sup>

<sup>1</sup> *Dep. of Computer and Software Engineering, Polytechnique Montréal, Montréal*

**Abstract**—**TODO: To writte after the paper is finished**

**Keywords**— Deep Learning, Convolutional Neural Network, Design Smells, Meta-Modeling, Software Engineering

## I. INTRODUCTION

L'apprentissage profond est un sous ensemble du machine learning qui utilise des réseaux de neurones artificiels pour apprendre des données non structurées. L'apprentissage profond est devenu très populaire ces dernières années grâce à ses performances dans plusieurs domaines comme la reconnaissance d'image, la reconnaissance vocale, la traduction automatique, etc. Les réseaux de neurones artificiels sont des modèles mathématiques qui sont inspirés du fonctionnement du cerveau humain. Ils sont composés de plusieurs couches de neurones qui sont connectés entre eux. Chaque neurone est composé d'une fonction d'activation qui permet de calculer la sortie du neurone en fonction de ses entrées. Il existe plusieurs types d'architecture de réseaux de neurones artificiels. Ses architecture ont été développées pour répondre à des problèmes spécifiques tels que l'architecture de convolution pour la reconnaissance d'image, les réseaux de neurones récurrents pour la reconnaissance vocale, etc. Dans ce papier, nous nous intéressons à l'architecture de convolution car elle permet de traiter un large ensemble de problèmes comme la reconnaissance d'image, la reconnaissance de texte, la reconnaissance de séquence, etc. Par ailleurs l'architecture de convolution fait partie des architectures feed-forward qui sont une classe de réseaux de neurones artificiels qui sont composés de plusieurs couches de neurones et qui ne contiennent pas de cycles. Les réseaux de neurones feed-forward sont les plus utilisés dans l'apprentissage profond.

Nous parlerons dans ce papier de programmes d'apprentissage profond pour désigner les programmes contenant les réseaux de neurones. Tout comme les programmes standard, les programmes d'apprentissage profond peuvent contenir des défauts. Nous nous intéressons dans ce papier aux défauts de conception car se sont des défauts introduits tôt dans le cycle de développement du logiciel et ils peuvent avoir un impact négatif conséquent sur la performance et la qualité du logiciel. En effet les défauts de conception sont des défauts qui sont introduits par le développeur lors de la phase de conception du logiciel.

- Définition des défauts de conception
- Définit les défauts de conception dans les programmes d'apprentissage profond
- Définition différence entre odeurs, bugs, erreurs et défauts

Méta modélisation des programmes d'apprentissage profond

- [RQ1] Comment détecter les défauts de conception dans les programmes d'apprentissage profond?
- [RQ2] Quels sont les défauts de conception les plus répandu dans

les programmes d'apprentissage profond CNN?

- [RQ3] Exist-il des lien entre les défauts de conception dans les programmes?

## II. STUDY DESIGN

*RQ1: What Ethereum smart contract code refactoring catalog is known to date?*

*RQ2: Does code refactoring change the gas price?*

*RQ3: What kinds of refactoring impact the gas price?*

### A. Data collection

Smart Bugs : <https://github.com/smartbugs/smartbugs-wild/tree/master/contracts>

### B. Data Processing

In this sub-section, we describe the details of the data collection and analysis approach followed to answer our different research questions. This approach is depicted in Figure 1. In the following, we elaborate on each data processing step.

- 1) *Collecter et trier les Smart contract:*
- 2) *Mesurer le prix du gaz de chaque Smart contrat:*  
<https://github.com/paperSubmission2020/GasmetReplicationPackage>
- 3) *Inserer des modifications (aléatoirement - manuellement):*
- 4) *Documenter les modifications:*
- 5) *Mesurer le prix du gaz des smart contrats modifiés:*

### C. Replication Package

## III. CASE STUDY RESULTS

**RQ1: What Ethereum smart contract code refactoring catalog is known to date?**

**RQ2: Does code refactoring change the gas price?**

**RQ3: What kinds of refactoring impact the gas price?**

## IV. DISCUSSION

## V. THREATS TO VALIDITY

This section discusses threats to the validity of our study following the guidelines for case study research.

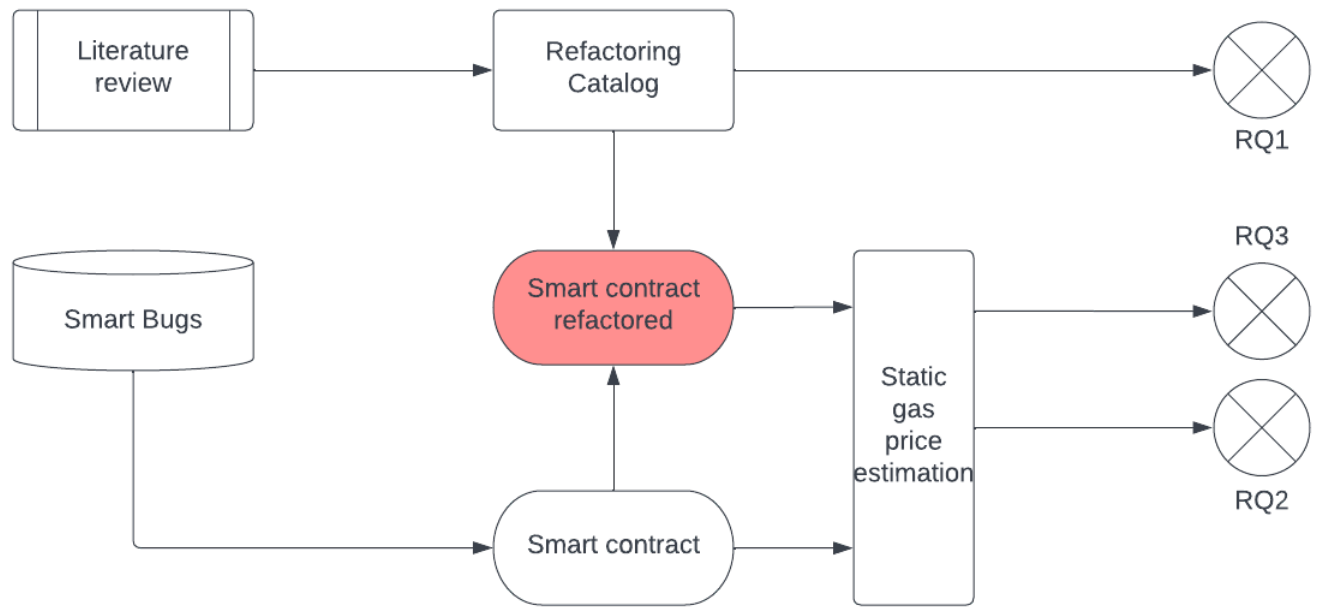


Figure 1: Overview of the study method

- A. Threats to construct validity
- B. Threats to internal validity
- C. Threats to External validity
- D. Threats to reliability validity

#### VI. RELATED WORK

- A. Replication studies in software engineering
- B. Studies on the limitations of SZZ
- C. Studies on the impact of Antipatterns

#### VII. CONCLUSION

[1]

#### REFERENCES

- [1] D. Jhon, "test," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 1082–1086.