УДК 004.413.2

MACHINE LEARNING MODEL FOR RECOGNIZING DESIGN PATTERNS

Asaad J. (ITMO university)
Supervisor–Associate Professor, Avksentieva E. Yu.
(ITMO university)

Introduction. Efficient software design is key to the reliability, scalability, and maintainability of systems. Design patterns provide reusable solutions to common design issues [1]. They help create robust and flexible architecture. Detecting these patterns in current codebases is vital. This is especially valid as software projects get more complex. This detection helps developers understand their current architecture. It allows them to check design quality and guide their refactoring efforts [2].

Main Part. This work assesses the role of an encoder transformer in software design pattern detection. We use new ideas in natural language processing and machine learning. First, we create a transformer-based model for feature extraction. Then, we train classifiers to find design patterns in source code.

Detecting design patterns has traditionally relied on manual analysis and expert insight. Developers often conduct a thorough examination of the code. They look at code structure, relationships, and behavior. They check if these align with known design pattern characteristics.

Even if advantageous, this experiential method is time-demanding, flawed, and substantially based on the knowledge and background of the analysts [3].

In this approach, we will first collect a dataset. Next, we will label, clean, and preprocess the data. Next, we will extract features with CodeT5+. Then, we'll train a machine learning classifier. This could be a Support Vector Machine (SVM), a Multi-Layer Perceptron (MLP), or another type. These classifiers will use the extracted features to identify design patterns in the code.

Conclusion Our study uses machine learning to uncover and classify design patterns. These smart techniques light the way in software engineering and improve program understanding. With better clarity, we make refactoring easier and streamline maintenance tasks.

References:

1. Talreja, D. and Sharma, R., 2024. Influence Of Gof On Software Stability A Case Study. Educational Administration: Theory and Practice, 30(5), pp.3651-3659
2. Asaad, J. and Avksentieva, E., 2024, April. A Review of Approaches to Detecting Software Design Patterns. In 2024 35th Conference of Open Innovations Association (FRUCT) (pp. 142-148). IEEE.
3. Nazar, N., Aleti, A. and Zheng, Y., 2022. Feature-based software design pattern detection. Journal of Systems and Software, 185, p.111179.