Hybrid Deep Learning-Based Classification of Type-1 and Type-2 Code Clones Across Multiple Programming Languages

H.K.G.S. Prabodhani[1]*, S. Vasanthapriyan[2], T.D. Rubasinghe[3]

[1]*Department of Computing & Information Systems, Faculty of Computing, Sabaragamuwa University of Sri Lanka, Belihuloya ,70140, Sri Lanka*
[2]*Department of Data Science, Faculty of Computing, Sabaragamuwa University of Sri Lanka, Belihuloya, 70140, Sri Lanka*
[3]*Department of Statistics and Computer Science, University of Peradeniya, Peradeniya, 20400, Sri Lanka*
**hkgsprabodhani@std.appsc.sab.ac.lk*

Code clone detection is a crucial software engineering task that supports software maintenance, code reuse, and defect prevention by identifying duplicated or highly similar source code fragments. However, detecting Type-1 and Type-2 code clones across multiple programming languages remains challenging due to variations in syntax, lexical characteristics, and structural representations. Existing clone detection techniques often exhibit limited cross-language adaptability as they are primarily designed for individual programming languages or depend on a single form of code representation. This research proposes a hybrid deep learning framework that integrates lexical and structural code representations for effective Type-1 and Type-2 code clone classification across Java, Python, and C++ programming languages. A quantitative empirical evaluation approach is adopted using benchmark datasets, including BigCloneBench, CodeXGLUE, and CodeSearchNet. The source code samples undergo preprocessing steps, including tokenization, normalization, identifier abstraction, class balancing, and duplicate removal, to improve data consistency and minimize potential data leakage. The datasets are partitioned into training, validation, and testing subsets using a consistent 70:15:15 ratio to ensure fair and reliable model evaluation. The proposed Hybrid CNN-BiLSTM-Attention framework integrates lexical token sequences with Abstract Syntax Tree (AST)-based structural representations. CNN layers extract local syntactic patterns, BiLSTM networks capture contextual dependencies, the Attention mechanism emphasizes significant code regions, and AST features provide complementary structural information. The framework is evaluated against baseline deep learning models, including CNN, LSTM, and Transformer-based approaches, using accuracy, precision, recall, F1-score, ROC-AUC, confusion matrix, and cross-language evaluation metrics. The experimental results demonstrate that the proposed framework achieves 92% accuracy, outperforming the evaluated baseline models. Cross-language evaluation, performed by training on two programming languages and testing on an unseen third language, achieves accuracies of 83.2%, 84.5%, and 82.8% for C++, Java, and Python test scenarios, respectively. These results demonstrate strong multilingual generalization capability and confirm that combining lexical and AST-based structural representations improves Type-1 and Type-2 code clone detection performance. The proposed framework provides a scalable and effective solution for automated multilingual code clone classification. This research contributes a hybrid representation-based approach that enhances cross-language code analysis, reduces code duplication, and supports intelligent software engineering practices.

Keywords: abstract syntax tree, code clone detection, cross-language code analysis, deep learning, software maintenance