DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5568

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A HYBRID TECHNIQUE FOR SOFTWARE CLONE DETECTION BY USING TOKEN BASED AND LINE BASED APPROACH

Sheetal
Department of Computer Science & Engg[1, 2]
Sri Sai College of Engg and Technology, Manawala
(Amritsar) Punjab (India)

Rimmy Chuchra*
Department of Computer Science & Engg[1, 2]
Sri Sai College of Engg and Technology, Manawala
(Amritsar) Punjab (India)

Abstract : The presence of same code more than one time degrades the quality of the software that outperforms software maintainability which is too difficult to manage. In this research paper, authors designed a new methodology which is termed as "A Hybrid Technique for software code clone detection by using Token Based and Line Based Approach" (HTSCCDUToL$_i$A) whose main purpose is to detect different types of clones viz. Type 1, Type 2 and Type 3 clones etc. This newly designed methodology working is based on two different types of software cloning approaches viz. Line based Approach and Token Based Approach. The major objective of this research paper is to remove redundant code or free space which is covered by the comment lines especially.The main significance to detect clones is to improve software code reusability and decrease same lines of source code. The major benefit to propose this new designed methodology is automatic detection of software code clone within minimum duration of time.In addition, the other benefit to design this new hybrid technique is to save software developer time, space as well as effort. By utilizing this new designed methodology the amount of code clones under a specific project or specific application can be easily reduced or removed up to some extent that will ultimately increase the performance of the software. In this way, this new designed methodology in future will helpful for producing more consistent or more efficient results.

Keywords: Software Cloning, Cloning techniques, Line Based Approach, Token Based Approach, Time, Space, Software Developer, Lines of Code (LOC), Redundant data.

I. INTRODUCTION

Due to the complex structure of software the understandability and maintainability becomes too difficult day by day. As the most common programming named "copy and paste programming" shows its negative impact. So, to avoid or minimize this copy and paste programming and improve the percentage of software reusability this software cloning concept is introduced.Software Code clone means presence of same code for more than one time. This will ultimately degrades the quality of the software.As studied by the authors there are different types of clone categories are available in the market viz. Type 1, Type 2, Type 3 and Type 4 etc. Each type is uniquely different from another type of clone whose definition is given below:-

- ✓ **Type 1:** It is a code clone which identical same except the comments.
- ✓ **Type 2:** It is code clones in which user defined names are changes like name of literals and function names.
- ✓ **Type 3:** It is code clones in which lines are added or deleted and lines are interchanged.
- ✓ **Type 4:** It is a code clone which is not created intentionally. These types of clones are created un-knowingly the presence of similar code. These are very difficult to detect.[8]

The summary of clone taxonomy can be shown in table.1 whose main purpose is to briefly categorize about each type of clone:-

Table 1. Summary of clone taxonomy.

Type 1	Type 2	Type 3	Type 4
Exact clone	Renamed clone	Near-miss clone	Structural clone
Structural clone	Parameterized clone	Gapped clone	Function clone
Function clone	Near-miss clone	Non-contiguous clone	Reordered clone
	Function clone	Reordered clone	Intertwined clone
		Structural clone	Semantic clone
		Function clone	

These above given four different types of clones are detected by utilizing different types of software clone detection techniques viz. Text based method, Token based method, Abstract Syntax tree based method, Program dependency graph based method, Metric based method and Line based method etc. Every software clone detection method has described below:-

- ✓ **Text Based Method (TeBM): -** Sequence of lines or strings is considered in this method. It is purely text based or following lexical approach for detecting clones towards structural elements which is written in some specific language viz. C++, JAVA, .Net, PHP. No transformation &normalization is performed on the source code. When this technique is practically applied then software developer directly enters that specific source code into clone detection process.

- ✓ **Abstract Syntax Tree (AST) Based Method: -**Here variable names and literal values are considered and can be represented in a tree structure. This method is work through a compiler generator which is used to generate an annotated parse tree (AST) and compares its sub-trees by characterization metrics based on a hash function through tree matching [8].The hash function enables one to do parameterized matching, to detect gapped clones and to identify clones of code portions in which some statements are reorderedthat can be shown as given below figure.1:-

```
if x + y then a := i; else foo; end if;
if p     then a := j; else foo; end if;
if q     then z := k; else bar; end if;
```

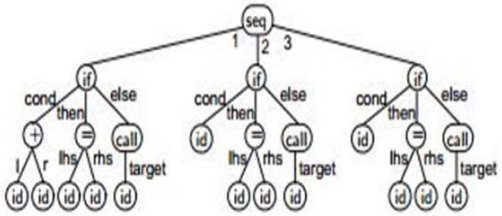

Figure 3: Sample Code and Abstract Syntax Tree of Code.

- ✓ **Program Dependency Graph (PDG) Based Method: -** This method is deals with data dependencies and control flow. Isomorphic sub-graph matching algorithm is applied in this Program Dependency Graph (PDG). Once the PDG is obtained from the source code [3] then clones can be easily detected. It is also helpful to detect non-contiguous code clones.

```
void bar() {
  int j = 1;
  int i = 0;
  while (j < 10)
    j++;
  printf("%d", i);
  printf("%d", j);
}
```

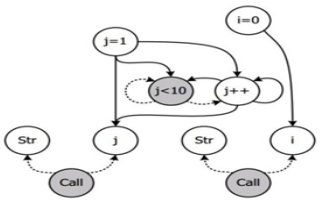

Figure 4: Sample code IN C and Dependency Graph.

- ✓ **Metric Based Method (MBM):-** Metrics are used to measure clones in software after the calculation of metrics from the original source code. This approach parses the source code to its AST/PDG methods representation for the calculation of metric [7]. Metrics can be easily calculated from the source code by utilizing several different tools as an example Columbus, Source monitor are available [6]
- ✓ **Token Based Method (TBM): -** At first step, this method uses parser or lexer for the transformation of source code into a sequence of tokens. In second step, the scanning of these sequences of tokens is done to find the same duplicate token sequences. In step third, the original code slices are represented by the token sequences will then be returned as clones.

- ✓ **Line Based Method (L_iBM):-** In line based method, code is matched with each line. In type 3 code clones, the lines of code are interchanged or lines are added or deleted. So, it is necessary to have some way to check code line by line rather than complete matching of code. In line based technique each line of first code is matched to each line of other code. Line Based technique has high accuracy.

In this research paper, authors proposes a new methodology which is termed as "A Hybrid Technique for software code clone detection by using Token Based and Line Based Approach" (HTSCCDUToL_iA). The working of this new designed methodology is based on Line based approach and Token based approach. The main purpose of this new designed methodology is to detect different types of clones viz. Type 1, Type 2 and Type 3 etc.The major objective of this research paper is to improve software reusability, software maintainability and software understandability. In addition, the benefit to utilize this new designed hybrid technique is to find software code clones automatically within minimum duration of time. As the data collected by the authors in their literature survey, by removing redundant code or duplicate code the software reusability can be easily improved.

II REVIEW OF LITERATURE

Heejo Lee and Hakjoo Oh et al (2017) [1]:- This paper proposes VUDDY. It is a type of approach which is used for the scalable detection of vulnerable code clones. The main benefit to propose this technique is it is more helpful for detecting security vulnerabilities in large software systems efficiently. This proposed method firstly pre-process billions of lines of code in 14 hours and 17 minutes after this it requires further few seconds to identify code clones. In addition, it also extends the scope of VUDDY to identify variants of known vulnerabilities with high accuracy.

Sukhpreet Kaur and Manpreet Kaur et al (April-June 2017) [2]:-This paper proposes two different types of investigations at first authors apply ant colony optimization technique which is used to generate optimized dataset and second investigation is to predict the code clones by using back propagation neural network classifier. Different software metrics as an example LOC, total function, Functions Repetitive, Public Variable which are generated from some JAVA programs used in MATLAB language. As authors study founded, Back Propagation Algorithm gives more accurate results with faster training and testing of Neural Network by considering various performance metrics as an example false acceptance rate(FAR), False Rejection Rate (FRR), Recall, Precision and accuracy etc.

Nguyen H.A et al. (2017)[3]:-This paper presented a tool JSync for clone management system is detected. This tool

provides support to clone detection and updation, clone change management, clone consistency, validating, clone synchronizing and clone merging. It represents abstract syntax tree which measures for code similarity. A new technique has been introduced which computes tree editing script, to detect and update clones of code. The study is done on open source systems which show that JSync is very efficient and accurate in clone detection and updating.

Zibran M.F et al. (2016)[4]:-In this paper, Authors works towards conflict-aware optimal scheduling of code clone refactoring is presented. To estimate the refactoring effort, an effort model is proposed for refactoring clones of code Object oriented and procedural source code. They captured the risks of refactoring in a priority scheme. They firstly refactor the Object oriented source code and their CP approach is a technique that no one else in the past reported to have applied in this context. Combining the strengths from both AI and OR, the CP approach has been shown to be effective in solving scheduling problems.

MeenaBharti and RajanGoyal et al (December 2014)[5] :- In this paper, authors discusses about repeatedly this clone finding activity degrades the quality of the software and hence these duplicate fragments automatically decrease the overall maintenance cost of that particular software.

III RESEARCH DESIGN

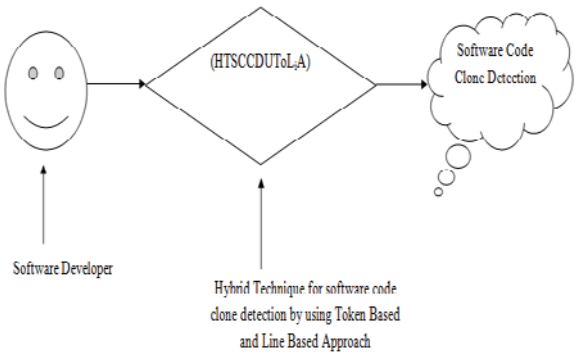

Figure 4: Interaction between Software Developer and Software.

IV PROPOSED METHODOLOGY (HTSCCDUTOL$_I$A)

Hybrid Technique for software code clone detection by using Token Based and Line Based Approach**)**

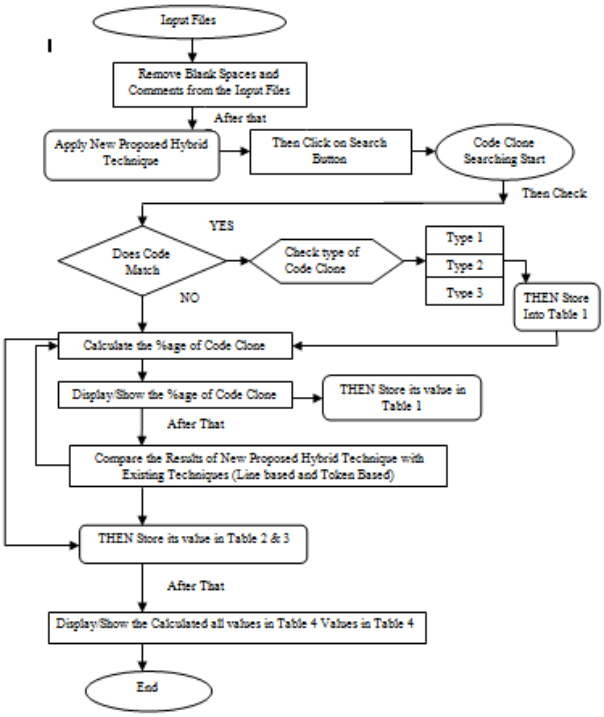

Figure 5: A Roadmap for Hybrid Technique for software code clone detection by using Token Based and Line Based Approach (HTSCCDUToL$_i$A).

V CONCLUSION

Different types of software clone detection techniques are discussed in this research paper.The main significance to study the concept of software cloning is to remove duplicate code or redundant code from any specific project or specific application. Authors designed a new methodology whichis termed as “A Hybrid Technique for software code clone detection by using Token Based and Line Based Approach” (HTSCCDUToL$_i$A) whose function is automatic code clone detection within minimum duration of time. This newly proposed hybrid technique is a combination of two approaches viz. Line based approach and Token based approach. The main function of this new designed methodology is to save application space, developer time as well as developer effort. By utilizing this newly designed Hybrid Technique three different types of clones are detected viz. Type 1, Type 2 and Type 3. For the detection of each type of code clone further different parameters are considered viz. Number of Clones detected Efficiency and portability etcand later on the results of different types of detected clones are stored in different tables. Hence, each one is considered as an important factor for detecting different type of clone.

VI REFERENCES

[1] Heejo Lee and Hakjoo Oh, “A Scalable Approach for Vulnerable Code Clone Discovery”, IEEE Symposium on Security & Privacy”, Korea University, 2017, Korea.

[2] Sukhpreet Kaur and Manpreet Kaur,” Code Clone Detection Using Metrics based Technique & Classification Using Neural Network”, International Journal of Research in

Electronics and Computer Engineering", Vol.5, Issue 2, April-June 2017.

[3] Nguyen H.A,"Clone Management for Evolving Software", IEEE transactions on software engineering, Vol.38, No.5, September-October 2017.

[4] Zibran M.F. and Roy C. K.,"Conflict-aware optimal scheduling of prioritized code clone refactoring", IET Software, Vol. 7, Issue 3,2016.

[5] MeenaBharti and RajanGoyal, "Software Cloning & its detection methods", Vol.5, Issue 4, December 2014.

[6] K. Rainer. F. Raimar, F. Pierre, "Clone Detection Using Abstract Syntax Suffix Trees", Working Conference on Reverse Engineering, 2006.

[6] K. Rainer.F. Raimar, F. Pierre, "Clone Detection Using Abstract Syntax Suffix Trees", Working Conference on Reverse Engineering, 2006.

[7] M. Hiroaki, H. Keisuke, "Folding Repeated Instructions for Improving Token-based Code Clone Detection",IEEE 12th International Working Conference on Source Code Analysis and Manipulation, Trento, 2012.

[8] H. Yoshiki, U. Yasushi, "Incremental Code Clone Detection: A PDG-based Approach", IEEE18th Working Conference on Reverse Engineering, 2011.

