

Comprehensively Meld Code Clone Identifier for Replicated Source Code Identification in Diverse Web Browsers

Dr. R. Dhaya,
Department of Computer Science,
Sarat Abida Campus -King Khalid University,
KSA,
dhayavel2005@gmail.com

Dr. R. Kanthavel,
Department of Computer Engineering,
King Khalid University,
Abha,
KSA
kanthavel2005@gmail.com

Abstract: The emergence and the progress in the process of reusing the software's, has caused difficulties in the maintaining the software codes and the corresponding depositories. Cloning of software codes is the important reason behind the arising difficulties in the maintenance of the software codes and the depositories. The cloning of codes is a process of replicating the existing codes for utilizing it elsewhere within the software system. The copying and pasting of the fragments of code is also well thought-out as the method of code cloning causing difficulties in the software maintenance. The maintenance of software is described as the alteration performed over the existing software on the completion of the development as well as the implementation process. Utilizing the maintenance process in the software the software companies deliver the improvements and the additional enhancements according to the working circumstance. The maintenance mainly focuses on removing the bugs and to repair the identified faults in the time of execution to enhance the performance. The work focused on the paper is mainly a comprehensive study over the prevailing tools laid out in the process of code-clone identification. The different techniques employed, the challenges incurred, the blunders made in the development, the enhanced refactoring challenges and efforts in the comprehension of codes are explored in the paper. From the study it was understood that the code clone identification using the web based tools are more advantageous than engaging algorithms in identifying the clones. So the a hybridized (meld) web based code clone identifier tool is engaged in the process of identifying the cloning of codes in diverse web browsers in exhibited in the paper. The exhibited tool equips a highly powerful tool for detecting the clones in a precise an efficient manner. The duplication of codes are often vulnerable and could be malicious. So the exhibited work in the future concentrates in developing an extended tool to identify the malicious codes to improvise the process of code clone identification making it concise and effective.

Keyword: Clone Identifiers, Web Based Tools, Code Maintenance, Depositories, and Web Browsers

1. Introduction

Cloning of codes is otherwise known as the duplicating of software codes for reusing it in program. This is done by the software programmer to reduce the difficulties in programming. Cloning helps the programmer to minimize his chores just by doing the copy paste instead of beginning the program from scratch. This process minimize the chores of the programmer, but raises issues on the software quality developed.

The mistakes in the code cloned propagates continuously throughout the program wherever the code is copied and pasted. So it becomes difficult to remove or alter the error occurred, and thus results in increased complications in maintaining a system and the software subsequently. This has become one of the vital issues addressed in the software industry as the maintaining a software is a highly expensive in the development of the software.

According to the multitudes of “research studies” “large scale-software development companies” are maintaining the prevailing software’s spending huge sum of amount for it. The maintenance is usually described as the alteration made in the software that is developed or updates included while the software is developed and removal of bugs to set right the identified faults in order to enhance its performance. The figure.1 below provides the categories of the clones prevailing.

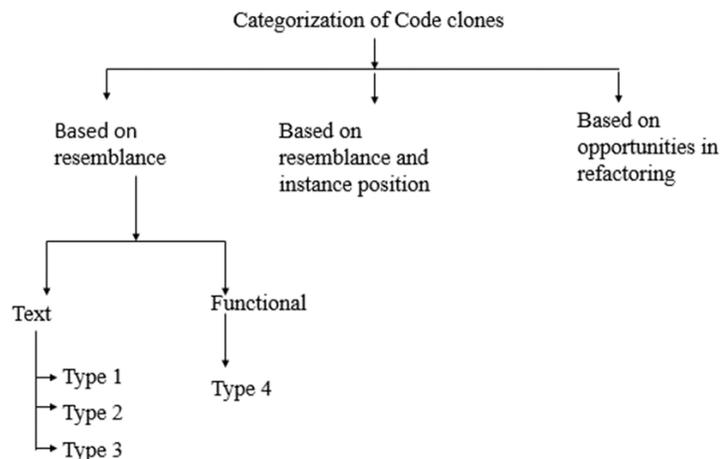


Figure.1 categorization of Cloned Codes [1]

The author Gautam, et al [1] has presented the categories of the code cloning as depicted in the figure .1 the different facets used as a basis to assort the cloned codes are i. The resemblance in the pair of code segments, ii. The instance position of the clone in the code, iii. The Opportunities in refactoring with the code replication. The resemblance in the codes are further categorized based on the identical contents and the functionalities. The identical contents are further categorized a type 1, 2 and 3 clones whereas the functionality based resemblance is holds subcategory named type 4. Types of clone based on the text and the functions are described in the figure.2 below.

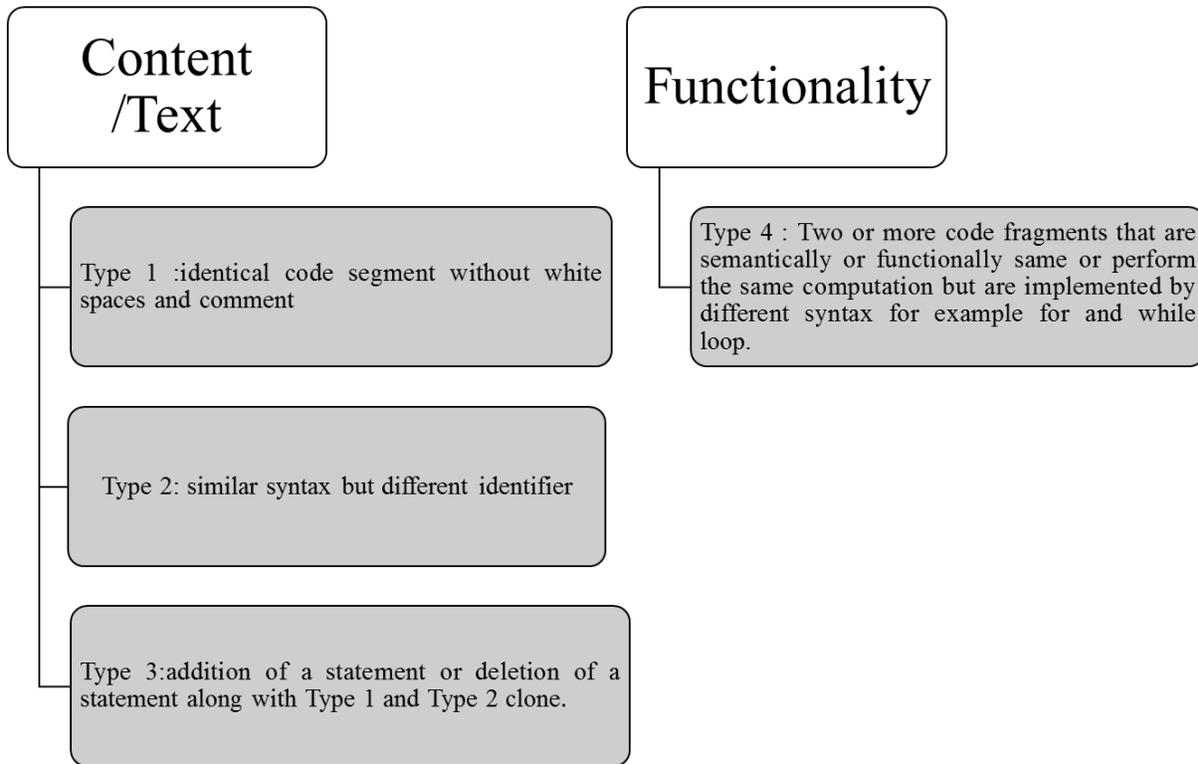


Figure.2 Types of Clones

According to the taxonomy of the clone the type of clones are distinguished as the exact, structural, functional, semantic etc. the table.1 presents the taxonomy of the clones.

Categories	Type 1	Type 2	Type 3	Type 4
Exact clone (similar code except some variations in comments)	✓			
Structural clone (it is based on level of similarity)	✓		✓	✓
Function clone (subset of structural)	✓	✓	✓	✓
Renamed clone (modification in copied code)		✓		
Parameterized clone (renamed clone with renaming)		✓		
Near-miss clone (slight modifications in copied fragment but syntactic structure remains same)		✓	✓	
Gapped clone (add, delete, modify some portion between segment)			✓	
Non-contiguous clone (Like near-miss, and gaps are allowed between code fragment)			✓	
Reordered clone (some statement have been reordered).			✓	✓
Intertwined clone (making two segments in to one segment)				✓
Semantic clone (semantically Similar codes)				✓

Table.1 Clone Taxonomy

The basic methodologies in identifying the cloned codes as summarized based on the insights gained from Kim, Miryung, et al [2], De Wit, et al [3] Jiang, et al [4] and Sharma, et al [5], some of the methodologies are “string or the textual based approaches, token based strategy, abstract syntax tree, program dependence graph, metric based, tracking clipboard, hash based and the locality sensitive hashing”, moreover the challenges in identifying the clone [6] are the “bug propagation, irregularities in change propagation, problem in system reform, design mistake, increased issues in refactoring and code comprehension effort”

Based on the researches done so far it was it was concluded that the type 3 and the type 4 clone are very difficult to identify thus demands highly efficacious and method and procedures to identify them. The identified clones can be erased manually, but this becomes tedious if the program hold thousands and thousands of lines. For this the web based tools were suggested, the repeating cloned code could even become dangerous. Regrettably as most of the available tools are less concise and incapable of providing a proper code identification the exhibited method in the paper aims in resolving the problem engaging a highly proficient web based tool for the identification of the clones. The exhibited method is devised with

the literature survey in section 2, the proposed powerful tool design in section.3 and the performance evaluation in section 4 and conclusion and the future work in section 5.

2. Literature Survey

Sheneamer, et al [7] has discussed the general types of clone codes identifying methods, challenges in it and tools essential for performing the identification. Bhalaji, N. et al [8] has performed the "Efficient and Secure Data Utilization in Mobile Edge Computing by Data Replication." Svajlenko, et al [9] has devised the "Cloneworks: A fast and flexible large-scale near-miss clone detection tool."

Pandian, A. Pasumpon, et al [10] has proposed the "Effective Fragmentation Minimization by Cloud Enabled Back up Storage." Ghofrani et al [11] has formulated a "conceptual frame work for identifying the clones using the machine learning" Raj, Jennifer S. et al [12] has presented an Efficient Information Maintenance Using Computational Intelligence in the Multi-Cloud Architecture." Smys, S., et al [13] elaborates the emerging trends in the "Computer Networks and Communication Technology" involved applications.

To reduce the time consumed in removing the cloned codes and delivery an efficient service the proposed method has exhibited a web based hybridized tool to identify the cloned codes in the source of diverse web browser.

3. Proposed Methodology

Segregating efficiently the types of clones would improve the eminence, heighten the reuse and minimize the overall cost of maintaining the software. The laid out methodology emphasis on employing hybridized (meld) technology to identify clone types. Web-centered –tool with automation is developed to identify the clones, the procedure put forth focuses on the identifying the type one, two, three and four clones and to design an equipment that could be easily setup, controlled remotely and capable of operating over diverse systems. The block diagram below in figure .3 depicts the hybridized cloned identifier (HCI).

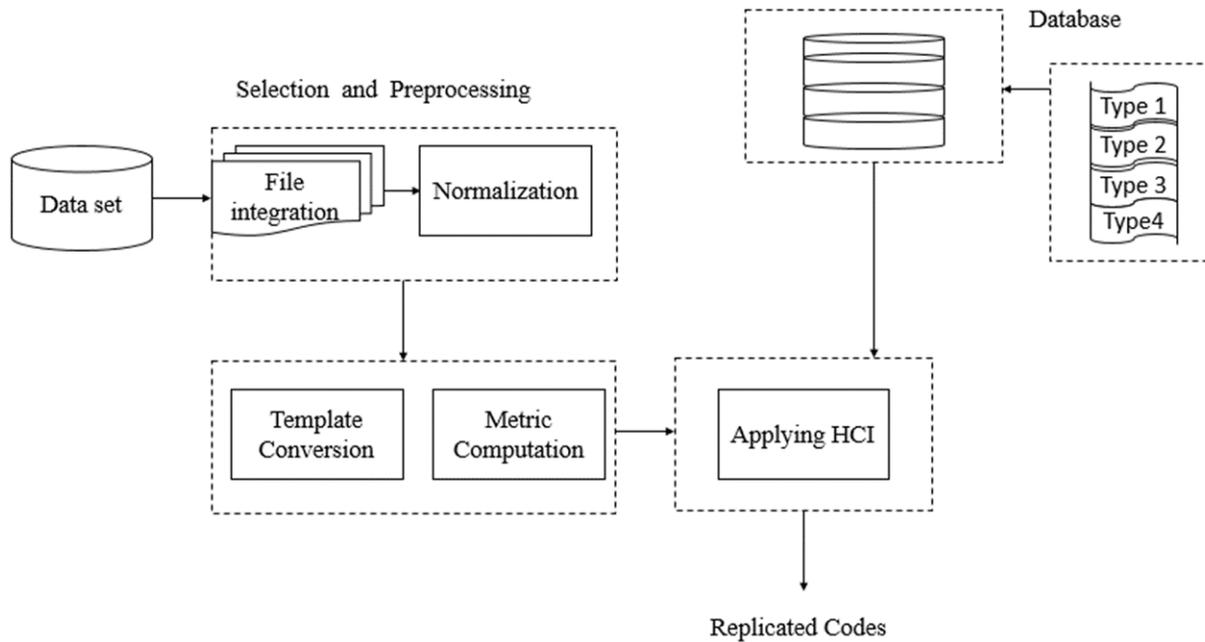


Figure.3 HCI Block Diagram

The procedure proposed in acquiring the particulars of clone in explained using two source code, the process is as shown in the figure.3 takes in samples of two codes in two different languages and the, code is preprocessed by subjecting to normalization, eliminating the additional spaces and the “header files”, the potential codes are extricated from the normalized codes and are preserved for the future orientations in processing the metrics and transforming the templates. In template transformation/conversion the source code takes the form of a “predefined set of statements or a standard intermediary form” and in the metric evaluation the count of lines in each function, the number of conditional statements, return , variable declaration and the looping statements are evaluated. Now applying the web based tool the types of cloning are identified, if the codes seems exact it is termed as type one and further checked for the next type i.e. type 2, to identify whether there are any “structural/syntactical similarity existing” if identified then it checks for the third type if there are modifications and still similar, if identified then checks for the fourth type whether codes with diverse fragments has same output. The identification of the clone types would provide a precise picture of the methods adopted in cloning the codes.

4. Performance Evaluation

The process of clone identification devised in the paper was tested with the prominent moderate size open-source web application, i.e. the Quick Auction, Web Wiz forms and the Snitz forms. All the above mentioned web applications are implemented using the MS ASP technology. So the number of web pages range from 10's to 100's the characteristics of the web applications are listed in the table.2

Web Applications	Version	Language Used	Number of Functions	Number of Files
Quick Auction	2.0.01	Visual Basic	40	52
Web-Wiz	3.4.075	Visual Basic ,JAVA	75	107
SNITZ	7.011	Visual Basic ,JAVA	124	237

Table.2 Characteristics of Web-Based Applications

Experiment output of the identified type of clones using the procedure emphasized in the compared with the “E-metric tool” developed by the “Bari University” the existing device examines the application adopting the MS ASP an observes the size of t applications with various granularity and choose the “homonym (equal identifiers utilized in the declaration script function as function names)-Program-Defined-Function” in the visual basic and the JAVA as the cloned operations. The range of clone types for the web based applications are depicted below in the figure. 4

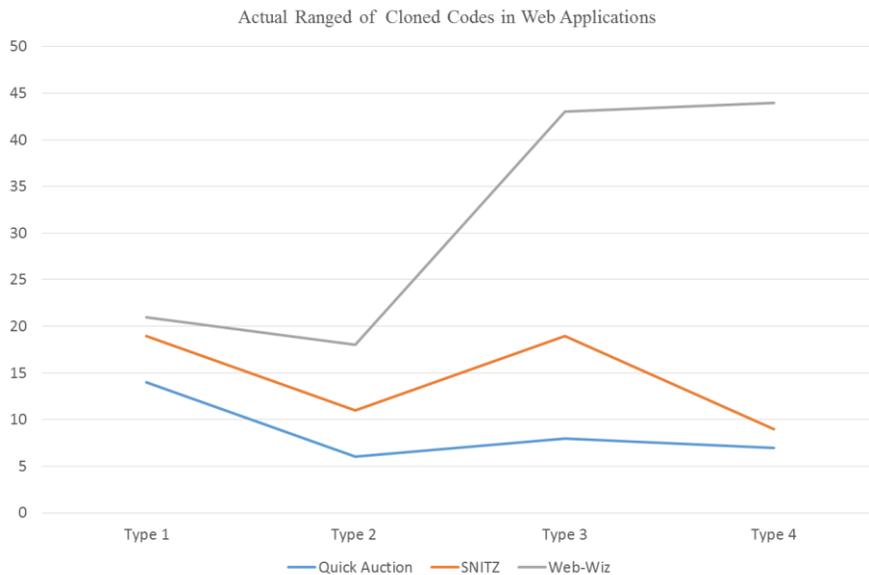


Figure .4 Range of Clone Codes in Web-Based Applications

The table.3 below is the clone type identified using the proposed and the existing “E-metrics” it was observed that “E-metrics” was less accurate as it was capable of identifying only 10 exact matches for type I and 2for type 2 (near miss functional) and 10 similar clones for i.e. type 3 and zero clones in type 4 in case of Quick auction and in case of Web wiz, “E-Metric” reported 25 , 9, 14 and 0 respectively for each type of clones and for SNITZ reported 20 , 4, 14 and 0 respectively for each type of clones.

Type of Clones	Quick Auction		SNITZ		Web-Wiz	
	HCI	E-metric	HCI	E-metric	HCI	E-metric
Type 1	15	10	19	20	26	25
Type 2	7	2	12	4	19	9
Type 3	9	10	20	14	44	14
Type 4	8	0	9	0	44	0
Precision %	96.5	89.9	82.45	60.57	91.05	78.78
Recall%	93	78.65	80.57	65.5	96.57	69.45

Table.3 Performance Comparison

From the results observed in table.3 it was concluded that the proposed method has established the desired output with the full automation achieving the desired percentage of precision and recall the figure.5 below depicts the results on the recall and precision.

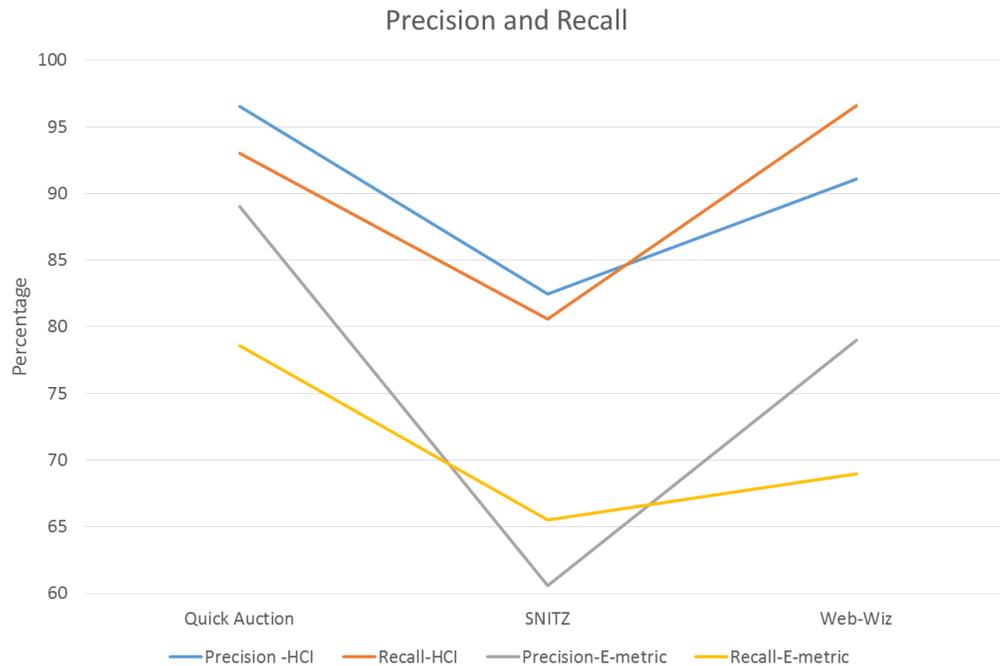


Figure.5 Recall and Precision

5. Conclusion

The clone identification procedure exhibited in the paper initiates with a comprehensive study on the different types and the taxonomy of cloned codes and also provides an overview on the challenges and the methods used in identifying the cloning. The study revealed that the type 3 and the type 4 clone are very difficult to identify thus demands highly efficacious and method and procedures to identify them. The identified clones can be erased manually, but this becomes tedious if the program hold thousands and thousands of lines. For this the web based tools were suggested, the repeating cloned code could even become dangerous. Regrettably as most of the available tools are less concise and incapable of providing a proper clone identification, the paper devised a web based tool to identify the replicated source code in the

diverse browsers, the tool was tested using three different web applications and was compared with the exiting tool. The results obtained concluded that the automated method put forth in the paper achieved the desired outcome with the determined level of precision as well as recall.

References

- [1] Gautam, Pratiksha, and Hemraj Saini. "Various code clone detection techniques and tools: a comprehensive survey." In *International Conference on Smart Trends for Information Technology and Computer Communications*, pp. 655-667. Springer, Singapore, 2016.
- [2] Kim, Miryung, Vibha Sazawal, David Notkin, and Gail Murphy. "An empirical study of code clone genealogies." In *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering*, pp. 187-196. 2005.
- [3] De Wit, Michiel, Andy Zaidman, and Arie Van Deursen. "Managing code clones using dynamic change tracking and resolution." In *2009 IEEE International Conference on Software Maintenance*, pp. 169-178. IEEE, 2009.
- [4] Jiang, Lingxiao, and Zhendong Su. "Automatic mining of functionally equivalent code fragments via random testing." In *Proceedings of the eighteenth international symposium on Software testing and analysis*, pp. 81-92. 2009.
- [5] Sharma, Yogita. "Hybrid technique for object oriented software clone detection." PhD diss., 2011.
- [6] Roy, Chanchal K., James R. Cordy, and Rainer Koschke. "Comparison and evaluation of code clone detection techniques and tools: A qualitative approach." *Science of computer programming* 74, no. 7 (2009): 470-495.
- [7] Sheneamer, Abdullah, and Jugal Kalita. "A survey of software clone detection techniques." *International Journal of Computer Applications* 137, no. 10 (2016): 1-21.
- [8] Bhalaji, N. "Efficient and Secure Data Utilization in Mobile Edge Computing By Data Replication." *Journal of ISMAC* 2, no. 01 (2020): 1-12.
- [9] Svajlenko, Jeffrey, and Chanchal K. Roy. "Cloneworks: A fast and flexible large-scale near-miss clone detection tool." In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pp. 177-179. IEEE, 2017.
- [10] Pandian, A. Pasumpon, and S. Smys. "Effective Fragmentation Minimization by Cloud Enabled Back Up Storage." *Journal of Ubiquitous Computing and Communication Technologies (UCCT)* 2, no. 01 (2020): 1-9.
- [11] Ghofrani, Javad, Mahdi Mohseni, and Arezoo Bozorgmehr. "A conceptual framework for clone detection using machine learning." In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 0810-0817. IEEE, 2017.

- [12] Raj, Jennifer S. "Efficient Information Maintenance Using Computational Intelligence In The Multi-Cloud Architecture." *Journal of Soft Computing Paradigm (JSCP)* 1, no. 02 (2019): 113-124.
- [13] Smys, S., Robert Bestak, Joy Iong-Zong Chen, and Ivan Kotuliak, eds. *International Conference on Computer Networks and Communication Technologies: ICCNCT 2018*. Vol. 15. Springer, 2018.

Authors Biography

Dr.R.Dhaya, works as professor in Department of Computer Science, at Sarat Abida Campus -King Khalid university, in KSA, his area of research are Artificial Intelligence, Data Management and Data Mining, Computer Architecture, Computer Networks, Robotics, Pattern Recognition, Computer Vision, Software Systems, Distributed Computing, quantum computers, Computer Graphics and Emerging Technologies

Dr.R.Kanthavel, professor in Department of Computer Engineering, in King Khalid University, Abha, in KSA his researchers are mainly focused on the emerging smart computing technologies that includes Distributed Computing, quantum computers, Computer Graphics, , Computer Networks, and Web Technologies