INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION & MANAGEMENT

IJRCM

A Monthly Double-Blind Peer Reviewed (Refereed/Juried) Open Access International e-Journal - Included in the International Serial Directories

Indexed & Listed at:

Ulrich's Periodicals Directory ©, ProQuest, U.S.A., EBSCO Publishing, U.S.A., Cabell's Directories of Publishing Opportunities, U.S.A., Google Scholar, Open J-Gage, India [link of the same is duly available at Inflibnet of University Grants Commission (U.G.C.)], Index Copernicus Publishers Panel, Poland with IC Value of 5.09 & number of libraries all around the world.

Circulated all over the world & Google has verified that scholars of more than **5555 Cities** in **190 countries/territories** are visiting our journal on regular basis.

Ground Floor, Building No. 1041-C-1, Devi Bhawan Bazar, JAGADHRI – 135 003, Yamunanagar, Haryana, INDIA

http://ijrcm.org.in/

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to the recent developments & practices in the areas of Computer Science & Applications; Commerce; Business; Finance; Marketing; Human Resource Management; General Management; Banking; Economics; Tourism Administration & Management; Education; Law; Library & Information Science; Defence & Strategic Studies; Electronic Science; Corporate Governance; Industrial Relations; and emerging paradigms in allied subjects like Accounting; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Rural Economics; Co-operation; Demography: Development Planning; Development Studies; Applied Economics; Development Economics; Business Economics; Monetary Policy; Public Policy Economics; Real Estate; Regional Economics; Political Science; Continuing Education; Labour Welfare; Philosophy; Psychology; Sociology; Tax Accounting; Advertising & Promotion Management; Management Information Systems (MIS); Business Law; Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labour Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; International Relations; Human Rights & Duties; Public Administration; Population Studies; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism & Hospitality; Transportation Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic; Web Design and emerging paradigms in allied subjects.

Anybody can submit the **soft copy** of unpublished novel; original; empirical and high quality **research work/manuscript** ***anytime*** in ***M.S. Word format*** after preparing the same as per our **GUIDELINES FOR SUBMISSION**; at our email address i.e. infoijrcm@gmail.com or online by clicking the link **online submission** as given on our website (***FOR ONLINE SUBMISSION, CLICK HERE***).

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____

THE EDITOR

IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF _____.**

(e.g. Finance/Mkt./HRM/General Mgt./Engineering/Economics/Computer/IT/ Education/Psychology/Law/Math/other, please specify)

DEAR SIR/MADAM

Please find my submission of manuscript titled '_____' for likely publication in one of your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published anywhere in any language fully or partly, nor it is under review for publication elsewhere.

I affirm that all the co-authors of this manuscript have seen the submitted version of the manuscript and have agreed to inclusion of their names as co-authors.

Also, if my/our manuscript is accepted, I agree to comply with the formalities as given on the website of the journal. The Journal has discretion to publish our contribution in any of its journals.

NAME OF CORRESPONDING AUTHOR	:
Designation/Post*	:
Institution/College/University with full address & Pin Code	:
Residential address with Pin Code	:
Mobile Number (s) with country ISD code	:
Is WhatsApp or Viber active on your above noted Mobile Number (Yes/No)	:
Landline Number (s) with country ISD code	:
E-mail Address	:
Alternate E-mail Address	:
Nationality	:

* i.e. Alumnus (Male Alumni), Alumna (Female Alumni), Student, Research Scholar (M. Phil), Research Scholar (Ph. D.), JRF, Research Assistant, Assistant Lecturer, Lecturer, Senior Lecturer, Junior Assistant Professor, Assistant Professor, Senior Assistant Professor, Co-ordinator, Reader, Associate Professor, Professor, Head, Vice-Principal, Dy. Director, Principal, Director, Dean, President, Vice Chancellor, Industry Designation **etc**. ***The qualification of author is not acceptable for the purpose***.

NOTES:

a) The whole manuscript has to be in ***ONE MS WORD FILE*** only, which will start from the covering letter, inside the manuscript. ***pdf. version is liable to be rejected without any consideration***.

b) The sender is required to mention the following in the **SUBJECT COLUMN of the mail**:

New Manuscript for Review in the area of (e.g. Finance/Marketing/HRM/General Mgt./Engineering/Economics/Computer/IT/ Education/Psychology/Law/Math/other, please specify)

c) There is no need to give any text in the body of the mail, except the cases where the author wishes to give any **specific message** w.r.t. to the manuscript.

d) The total size of the file containing the manuscript is expected to be below **1000 KB**.

e) Only the **Abstract will not be considered for review** and the author is required to submit the **complete manuscript** in the first instance.

f) ***The journal gives acknowledgement w.r.t. the receipt of every email within twenty-four hours*** and in case of non-receipt of acknowledgment from the journal, w.r.t. the submission of the manuscript, within two days of its submission, the corresponding author is required to demand for the same by sending a separate mail to the journal.

g) The author (s) name or details should not appear anywhere on the body of the manuscript, except on the covering letter and the cover page of the manuscript, in the manner as mentioned in the guidelines.

2. **MANUSCRIPT TITLE:** The title of the paper should be typed in **bold letters**, **centered** and **fully capitalised**.
3. **AUTHOR NAME (S) & AFFILIATIONS:** Author (s) **name**, **designation**, **affiliation** (s), **address**, **mobile/landline number** (s), and **email/alternate email address** should be given underneath the title.
4. **ACKNOWLEDGMENTS:** Acknowledgements can be given to reviewers, guides, funding institutions, etc., if any.
5. **ABSTRACT:** Abstract should be in **fully Italic printing**, ranging between **150** to **300 words**. The abstract must be informative and elucidating the background, aims, methods, results & conclusion in a **SINGLE PARA**. ***Abbreviations must be mentioned in full***.
6. **KEYWORDS:** Abstract must be followed by a list of keywords, subject to the maximum of **five**. These should be arranged in alphabetic order separated by commas and full stop at the end. All words of the keywords, including the first one should be in small letters, except special words e.g. name of the Countries, abbreviations etc.
7. **JEL CODE:** Provide the appropriate Journal of Economic Literature Classification System code (s). JEL codes are available at www.aea-web.org/econlit/jelCodes.php. However, mentioning of JEL Code is not mandatory.
8. **MANUSCRIPT:** Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. *It should be free from any errors i.e.* ***grammatical***, ***spelling*** *or* ***punctuation.*** *It must be thoroughly edited at your end*.
9. **HEADINGS:** All the headings must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.
10. **SUB-HEADINGS:** All the sub-headings must be bold-faced, aligned left and fully capitalised.
11. **MAIN TEXT:**

THE MAIN TEXT SHOULD FOLLOW THE FOLLOWING SEQUENCE:

INTRODUCTION

REVIEW OF LITERATURE

NEED/IMPORTANCE OF THE STUDY

STATEMENT OF THE PROBLEM

OBJECTIVES

HYPOTHESIS (ES)

RESEARCH METHODOLOGY

RESULTS & DISCUSSION

FINDINGS

RECOMMENDATIONS/SUGGESTIONS

CONCLUSIONS

LIMITATIONS

SCOPE FOR FURTHER RESEARCH

REFERENCES

APPENDIX/ANNEXURE

The manuscript should preferably be in ***2000 to 5000 WORDS***, But the limits can vary depending on the nature of the manuscript.

12. **FIGURES & TABLES:** These should be simple, crystal **CLEAR**, **centered**, **separately numbered** & self-explained, and the **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. *It should be ensured that the tables/figures are referred to from the main text*.

13. **EQUATIONS/FORMULAE:** These should be consecutively numbered in parenthesis, left aligned with equation/formulae number placed at the right. The equation editor provided with standard versions of Microsoft Word may be utilised. If any other equation editor is utilised, author must confirm that these equations may be viewed and edited in versions of Microsoft Office that does not have the editor.

14. **ACRONYMS:** These should not be used in the abstract. The use of acronyms is elsewhere is acceptable. Acronyms should be defined on its first use in each section e.g. Reserve Bank of India (RBI). Acronyms should be redefined on first use in subsequent sections.

15. **REFERENCES:** The list of all references should be alphabetically arranged. ***The author (s) should mention only the actually utilised references in the preparation of manuscript*** and they may follow Harvard Style of Referencing. **Also check to ensure that everything that you are including in the reference section is duly cited in the paper**. The author (s) are supposed to follow the references as per the following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc., in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italic printing. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parenthesis.
- ***Headers, footers, endnotes*** *and* ***footnotes*** *should* ***not be used*** *in the document.* However, **you can mention short notes to elucidate some specific point**, which may be placed in number orders before the references.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio" Ohio State University, Nigeria.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg, Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–23

UNPUBLISHED DISSERTATIONS

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITES

- Garg, Bhavet (2011): Towards a New Gas Policy, Political Weekly, Viewed on January 01, 2012 http://epw.in/user/viewabstract.jsp

ANALYSIS OF CODE CLONE DETECTION OF WEB LANGUAGE USING SUFFIX ARRAY BASED TOKENIZATION

GURVINDER SINGH
RESEARCH SCHOLAR
I. K. GUJRAL PUNJAB TECHNICAL UNIVERSITY
JALANDHAR

JAHID ALI
DIRECTOR
SRI SAI IQBAL COLLEGE OF MANAGEMENT AND IT
BADHANI

ABSTRACT

In recent time, code clones have gotten much consideration from numerous analysts in the field of software engineering. Code cloning of source code is a noteworthy issue for substantial, industrial systems. The real outcome of cloning is that it risks the maintenance procedure. Cloning is the most essential methods for software reuse. Code cloning has been widely utilized inside the product advancement outline group. A software clone is a code piece indistinguishable or like another in the source code. Clones are viewed as unsafe for programming support and evolution, since it expands complexity of the framework. In the event that a code is altered, it should be checked against all related code clones to check whether they should be modified also. Removal, avoidance or refactoring of cloned code is other essential issues in software maintenance. Detection of software clones minimizes the cost maintenance of software and furthermore builds understand ability of the framework. Many code clone detection procedures have been proposed to distinguish clones. Moreover, we propose a novel method and tool for clone location. The clones are distinguished by utilizing hybrid approach which is utilized to identify clones in different programming languages. The novel part of the approach takes the advantage of metric based and token based methods for clone identification. In the experiment, we compared the proposed tool with the existing tool with performance metrics such as Precision, recall and accuracy. Thus, we affirmed that the proposed instrument was better than the existing tool in simulation measures, which are appeared by utilizing Net Beans.

KEYWORDS

code clone, clone detection, hybrid approach, software metrics, suffix array.

INTRODUCTION

A numerous types of applications created today rely on object-oriented programming (OOP) languages. OOPs are today the predominant worldview in standard programming improvement. As their prerequisite is expanding step by step they are getting to be noticeably bigger and complex. Large scale programming frameworks are costly to build and, are considerably more costly to keep up. A code clone is a code portion in source files that is identical or similar to another [1]. A code clone is a code block in source files which is identical or similar to another code block. Code cloning or the act of copying code fragments and making minor, non–functional alterations, is a well-known problem for evolving software systems leading to duplicated code fragments or code clones. The normal functioning of the system is not affected, but further development may become prohibitively expensive. Cloning mainly occurs because programmers find that it is cheaper and quicker to use the copy and paste feature than writing the code from scratch. Sometimes programmers intent on implementing new functionality find some working code that performs a computation nearly identical to the one desired, copy it entirely and then modify in place [3]. Code cloning, is considered a serious problem in industrial software [2]. Duplicated code proves easy and cheap during the software development phase, but it makes software maintenance much harder. Software clone has a number of negative effects on the quality of the software. Besides increasing the amount of the code, which needs to be maintained, it also increases the bug probability.

TYPES OF CLONE

According to the different similarities, clone can be classified into two categories: One type of similarity considers textual similarity and other second considers the semantic level, which the clone code must have the same behaviors, means the functional similarity.

TEXTUAL SIMILARITY

Two code sections can be comparative in view of the likeness of their program content. The accompanying sorts of clones are examined keeping in mind the end goal to discover textual similarity [3].

- **TYPE I:** In Type I clone, a replicated code piece is the same as the first. Nonetheless, code pieces that is indistinguishable with the exception of changes in whitespace and remarks.
- **TYPE II:** A Type II clone is a code piece that is the same as the first aside from some conceivable variations about the comparing names of user-defined identifiers types, design and remarks.
- **TYPE III:** Copied parts with changes. Statements can be altered, added or removed notwithstanding varieties in identifiers, literals, types, layout and comments are called type III Clones.

FUNCTIONAL SIMILARITY

Two code parts can be comparative in light of the comparability of their functionalities without being textually comparable. On the off chance that the functionalities of the two code sections are indistinguishable or comparable i.e., they have comparative pre and post conditions alluded as Type IV clones [4].

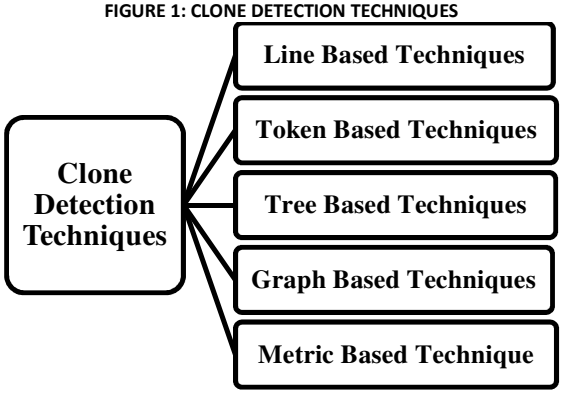
- **TYPE IV:** Type IV clones are the consequences of semantic similitude between at least two code sections. At least two code parts that play out a similar capacity however are executed through various language structure are called type IV clones.

CLONE DETECTION TECHNIQUES

Clone detection is an important topic in the research of clones. Many techniques, including those listed below, detect clones automatically. Clone detection techniques can be categorized into the following categories.

- Line Based Techniques
- Token Based Techniques
- Tree Based Techniques
- Graph Based Techniques
- Metric Based Techniques

FIGURE 1: CLONE DETECTION TECHNIQUES



- **Line Based Technique:** Line-based methods distinguish clones by looking at each line of code parts as a string [4]. Line-based systems can recognize clones rapidly as contrasted and other discovery strategies, since they don't require any pre-processing of the source code.
- **Token Based Technique:** Token-based systems change source code into a token grouping [5]. At that point, they identify regular sub-groupings of the tokens as clones.
- **Tree Based Technique:** In tree based system, comparative sub trees are searched in the tree. At the point when a match is discovered relating source code of the comparative sub trees are returned as clone combines or clone classes [6].
- **Graph Based Technique:** In diagram based methods, source code is changed into a chart portrayal [7]. A program dependence graph (PDG), which is one of the notable diagram portrayals.
- **Metric Based Technique:** Metric based systems figure different measurements for each program module, for example, method, function, and class and afterward recognize clones by contrasting the likeness of the modules in view of the measurements [8].

RELATED WORK

In Past, a broad number of studies have been finished with respect to numerous approaches for clone.

C.K.Roy ed.s [9] compared the various techniques of clone detection such as textual approach, lexical approach, semantic approach and metric based approach and furthermore comparing and evaluating clone detection tools such as Duploc, simian and NICAD. They proposed that NICAD tool is the best among all others. They use the clone detection process- pre-processing, transformation, match detection, formatting, filtering and aggregation. They also provide examples of how one may utilize the results of this examination to pick the most proper clone discovery apparatus or strategy with regards to a specific arrangement of objectives and limitations.

D.G.Devi ed.s [10] detects the different types of clones using different algorithm like textual analysis, metric based distance algorithm and mapping algorithm. The detected clones are-extract clone, renamed clone, gapped cloned and semantic clone. They utilized clone identification and metrics to assess quality. In situation based algorithm the two strategies are utilized –token based and line based. A token-based technique will be more costly as far as time and space complexity than a line-based one because a source line contains several tokens. Metric based clone detection approach uses the metric based distance algorithm. Then they compared different types of approach using different algorithm and calculate their metrics, speed, cost and quality.

A. De Lucia ed.s [11] illustrate a novel approach for updating Web Applications(WAs) in view of clone analysis that objective at identifying and generalizing static and dynamic pages and navigational examples of a web application. Clone analysis is likewise useful for distinguishing literals that can be produced from a database. A case study is described which indicated how the proposed approach can be utilized for restructuring the navigational structure of a Web Application by removing repetitive code. An instrument to identify and analyze cloned patterns in web applications utilizing clone analysis and clustering of static and dynamic web pages. The tool has been implemented for WAs developed using PHP or JSP technology. It supports the user to filter out details that do not contribute to the analysis of cloned patterns.

G.Gupta ed.s [12] plan and implement a Code Clone Detector instrument to detect clones. The novel aspect of the work is done by utilizing metric based approach on Java source codes. For calculating metrics Java byte code is utilized and after that source code refactoring is done in order to minimizing code clones. Since the byte code is taken which converts the source code into uniform representation and it is given as an input to the tool for calculating metrics value, so up to some extent it is able to identify the semantic clones. Also byte code is platform independent which makes this tool more productive than the already existing tools.

T.Muhammad ed.s [13] displayed Dynamic web pages composed of inter-woven (tangled) source code written in various programming languages (e.g., HTML, PHP, JavaScript, CSS) makes it hard to investigate and manage clones in web applications. They display an exploratory investigation on the patterns of both exact (Type-1) and near-miss (Type-2 and Type- 3) code clones in two industrial web applications, which underwent two different development styles. One was developed using the traditional style where HTML mark-up and PHP code were put together on dynamic web pages. The other was developed following a more sophisticated approach using the MVC (Model-View-Controller) pattern that resulted in a relatively more modularized implementation.

RubalaSivakumar ed.s [14] identified that the code cloning is not only found in software products and programming language but it can also be found in web applications. Scripting languages such as ASP, JSP, PHP etc are used in the development of web sites in which code duplication practice usually involved in making of several web pages. The proposed tool gives its evaluated result in precision and recall parameter which then further compared with the other existing. The result of comparison showed that the value of precision and recall in term of percentage with the proposed tool using.NET gives higher value with accuracy than the other tool.

S.Ur. Rehman ed.s [15] demonstrates a new code clone detection technique. A method is capable for distinguish clones inside vast source codes and is particular in its capacity to detect code duplication independent of the source language. They are additionally working on some of its future directions including the removal of the clones detected from the source code.

OBJECTIVES OF THE STUDY

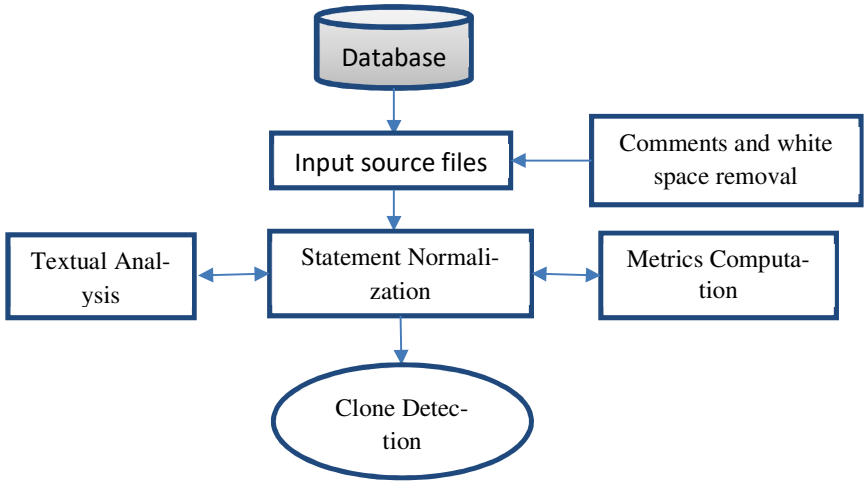
1. Analysis of code clone detection of web language using suffix array based tokenization.
2. To evaluate result to compare with existing technique.

PRESENT WORK

Cloned software is a noteworthy issue in extensive software frameworks that have advanced over along period of time. The issue is not amenable to a straight forward solution to totally reengineer the framework is a conceivable option, but is very cumbersome and involves much cost. Current work presents a practical method, for detecting exact clones for object oriented program source code. In this model a hybrid approach using combination of metrics based and token based technique is used for detection of clone. The approach is divided into two stages. In the primary segment, metric primarily based method is used to hit upon the capability clones. The metrics are calculated primarily based on class degree and characteristic degree and the edge stage is also defined for the matching of

metrics. In the second phase, if the metrics healthy count reaches the edge fee then simplest actual clones are detected through using Suffix array based tokenization technique.

FIGURE 2: CLONE DETECTION ARCHITECTURE



PROPOSED APPROACH INCLUDED FOLLOWING STEPS

Step 1. Input two source files related to web language like asp.net and php.
Step 2. Pre-processing and Filtering of file by removing white spaces and all types of comments.
Step 3. Metric Calculation
List of metrics (Function & Method level)
Class level metrics

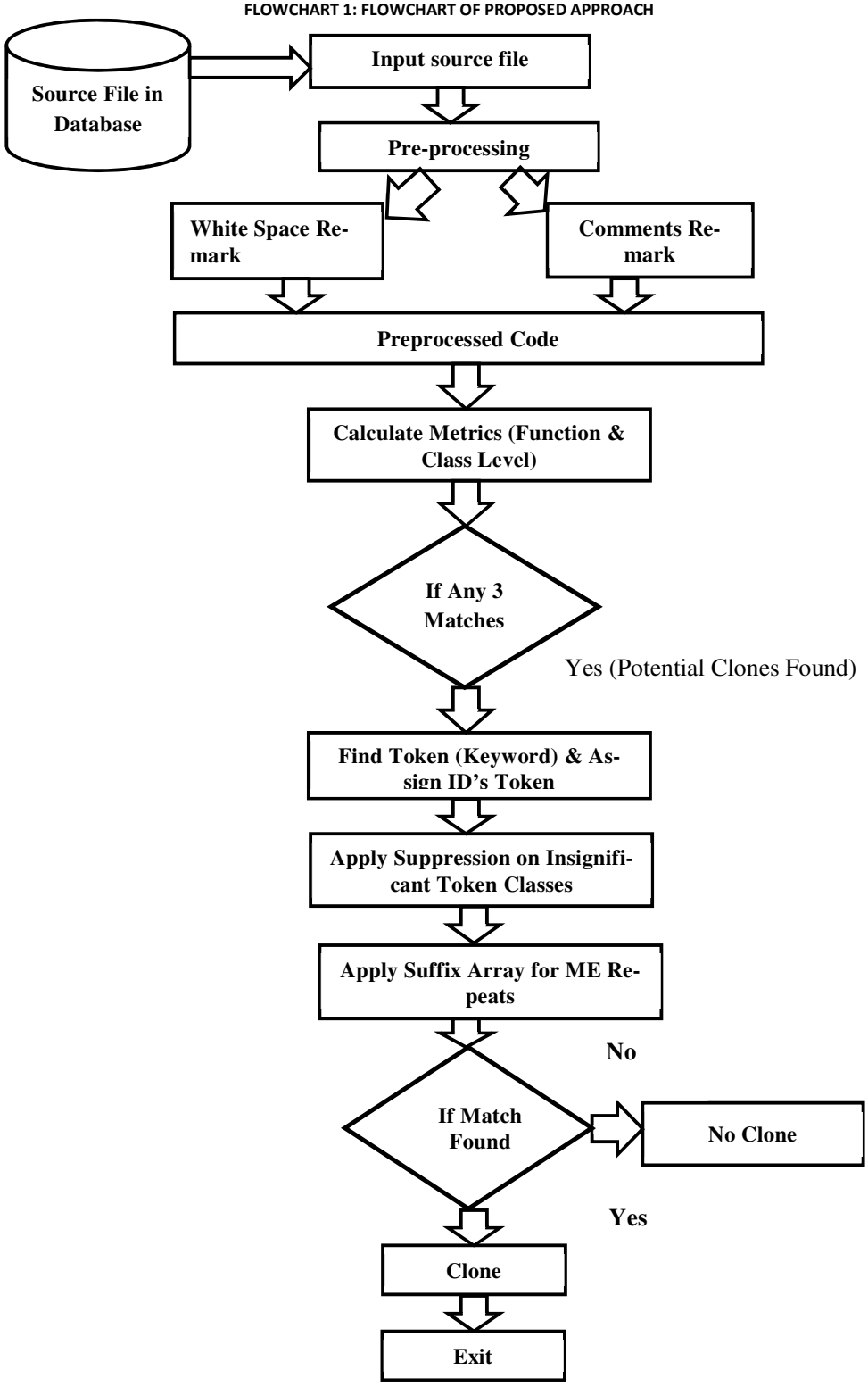
- LOC
- No of private variable
- No of public variable
- No of protected variable
- Total no of variable
- No of loop control statements(for+ while+ do-while)
- Redirect statements
- No of conditional statements (if + switch)
- Friend variable
- Private function
- Public function
- Protected function
- Total no of functions

Function level metrics

- Function name
- No of local variables
- No of Function calls
- No of arguments/parameters passed in function
- No of loop controls
- No of return statements.

Step 4: Calculating the tokens and detection of clones using Suffix array based tokenization technique.
Step 5: Analysing the results

FLOWCHART 1: FLOWCHART OF PROPOSED APPROACH



RTF TOKENIZATION MECHANISM

Repeated Tokenization Finder (RTF) likewise enables the client to tailor the token string for better clone detection. The principal probability is the suppression of certain insignificant token classes that may be considered noises in source code for clone recognition. For instance, contrast of get to modifiers like private, protected, public and so forth may not be exceptionally intriguing between two generally indistinguishable techniques and subsequently these tokens can without much of a stretch be stifled. The client can basically demonstrate, from a rundown of all dialect tokens, the tokens that ought to be stifled amid code checking and these tokens won't turn into a piece of the last token string.

TABLE 1: SIMPLY TOKENIZATION SCHEME

TOKEN CLASS	ID
Keywords	
private	1
public	2
protected	3
return	4
...	...
Operators	
+	31
*	32
...	...
Identifiers	40
Constants	41
Literal strings	42
Punctuation symbols	
(	51
)	52
{	53
}	54
;	55
,	56
...	...
Type names	
int	71
short	72
...	..

Another alternative offered is to compare the distinctive token classes. For instance, if the client does not have any desire to separate between the sorts {int, short, long, float, double}, a similar ID can be utilized to speak to each individual from the above arrangement of sorts. Along these lines, every one of those code parts that vary just in the kind of specific factors end up noticeably correct copies of each other in the token string. Clients can likewise liken operators, certain keywords, punctuation symbols, access specifiers and so on. Contingent on the necessity.

1 7140 5171 40 5671 40 52 53 4 4031 55 542 72 40 5172 405672 40 52 53 4 4032 55 54

The repetitive parts of the string are underlined. On the off chance that the client chooses to smother tokens private (token ID 1), open (token ID 2) and secured (token ID 3), the resultant token string would progress toward becoming:

71 40 51 71 40 56 71 40 52 53 4 40 31 55 54 72 40 51 72 40 5672 40 52 53 4 40 32 55 54

Presently, if the operators + (token ID 31) and * (token ID 32) are likened, and furthermore sort names *int* (token ID 71) and *short* (tokenID 72) are compared, we would get only two indistinguishable substrings covering the two procedures absolutely, plainly demonstrating the abnormal state of similitude introduce between the two system:

71 40 51 71 40 56 71 40 52 53 4 40 31 55 5471 40 51 71 40 5671 40 52 53 4 40 31 55 54

It ought to however be noticed that utilizing these components of controlling the input tokens discretionary and does not hamper the ease of use of the instrument. Rather, it is additional preferred standpoint for advanced users, a classification that we can securely expect that maintenance engineers should fall into. RTF can successfully distinguish helpful clones even with no any user-defined customizations. However, if the users suspect, based on their experience with the source code, that by customizing the input tokens they can show signs of improvement comes about, RTF gives the flexibility to do so.

Clone detection is basically an issue of looking repeated and non-overlapping substrings which are the regular prefix of some suffixes. Along these lines, the key of solving this problem is to use suffix array. The suffix array (SA) is an array SA[0..*n*-1] in which SA[*j*] = *i* if suffix *i* is the *jth* in lexicographical order among all the suffixes of *S*. The suffix array of a string of length *n* over an integer alphabet can be computed in O (*n*) time. After building the suffix array, we simply utilize some programming techniques to locate every repeated substring whose lengths are in the interval [max, min].

FIGURE 3: SA AND LCP ARRAYS OF STRING *S*

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S	A	T	G	C	A	A	T	G	C	C	V	G	G	C	A	T	T	G	C	A	T	V
SA	4	0	5	14	19	3	13	18	8	9	2	12	17	7	11	1	16	6	15	20	10	21
LCP	-1	1	4	2	2	0	2	3	1	1	0	3	4	2	1	0	4	3	1	1	0	0

Give us a chance to mean the length of the longest common prefix of suffixes i and j by LCP (i, j). At that point, the LCP array contains the lengths of the longest common prefixes between successive suffixes of SA. That is, LCP[i] = lcp(SA[i-1], SA[i]) for $0 < i \leq n-1$. Given S and SA, LCP can also be computed in θ (n) time [21]. An example of this is shown in Figure 3.

ALGORITHM 1: PSEUDO CODE OF CRFINDER ALGORITHM

```
CRFinder(S, Pmin, Fmin)
Input: string S of length n, requested minimum repeat length threshold Pmin and minimum frequency Fmin
Output: all complete repeats (p; i, j) in S that appear at least Fmin times and period p ≥ Pmin
Preprocessing: Computer SA[i] and LCP[i] (0≤i≤n-1) of string S; let LCP[0] = -1, LCP[n] = -1
1. k=0; push (STALOCH; 0, 0)
//(location, height) are stored in stack STALOCH, initial values set to 0
2. while (k <n-1)
3. while (LCP[k] <= LCP[k+1]) do
//when LCP[k]<LCP[k+1]: open a potential repeat occurring
4. if (STALOCH[top].height < LCP[k+1]) then
push (STALOCH; k, LCP[k+1])
5. k++
6. while (LCP[k] > LCP[k+1]) do
//when LCP[k] > LCP[k+1]: close the repeat previously created
7. j=k
8. k++
9. pop(STALOCH) to (i,h)
10. while (STALOCH[top].height > LCP[k]) do
11. for p = h down to STALOCH[top].height +1 do
12. Check(p, i, j)
//check period and occurrence frequency satisfy the requirements
13. pop(STALOCH) to (i,h)
14. for p = h down to LCP[k] +1 do
15. Check(p, i, j)
16. if (STALOCH[top].height < LCP[k]) then
push (STALOCH; i, LCP[k])
17. while (LCP[k] = LCP[k+1]) do
//when LCP[k] = LCP[k+1]: sustain a potential repeat occurring
18. k++
Function Check(p, i, j)
1. f = j-i+1
2. if f >=Fmin and p >= Pmin then
3. output (p; i, j; f)
```

RESULTS AND DISCUSSIONS

This section exhibits the simulation results of the proposed technique. We study that our proposed method detects the clones available within the supply documents in an efficient way. We have compared the proposed technique with the already present clone detection device supply monitor as a way to deliver less precision and take into account charge while as compared to our proposed technique. We have done trials with the existing and proposed system to recognize the clones. The after effects of the methodology have been tried on two web source code records and their resultants are appearing in table 2 and table 3.

EVALUATION PARAMETERS

The evaluation parameters include various terms such as true positives, true negatives, false negatives and false positives. These are the terms that are used to compare the class labels assigned to documents with the classes the items actually belong to by a classifier. True positive terms are truly classified as positive terms. False positive are not labeled by the classifier as positive class but should have been. True negative terms are correctly labeled as in negative class by the classifier. False negative terms are those terms that are not labeled by the classifier as belonging to negative class but should have be classified. Confusion Matrix contains these terms that are used for evaluation.

CONTINGENCY TABLE

		Correct Labels	
		Positive	**Negative**
Classified Labels	**Positive**	True Positive ($n_{s\to s}$)	False Positive($n_{h\to s}$)
	Negative	False Negative($n_{s\to h}$)	True Negative ($n_{h\to h}$)

Precision, recall and accuracy parameters are used for evaluating the performance:

PRECISION AND RECALL

It verifies what number of cloned documents is effectively identified as clones among those all that are having clones. It can be measured by number of files cloned that are effectively recognized as clone to the aggregate number of files taken as clones.

Precision = $\frac{n_{s\to s}}{n_{s\to s}+n_{h\to s}}$ Eq. (1)

Recall = $\frac{n_{s\to s}}{n_{s\to s}+\boldsymbol{n_{s\to h}}}$ Eq. (2)

ACCURACY

Accuracy is the percentage of effectively identified files as clone. It can be measured as the quantity of correctly classified files as clones to the aggregate number of files. It should be highest for the best technique.

Accuracy = $\frac{n_{s\to s}+n_{h\to h}}{n_{s\to s}+n_{s\to h}+n_{h\to s}+n_{h\to h}}$ Eq. (3)

We have done trials on 50 files from which 40 files are actual cloned files and 10 are non-cloned files. A few files are taken from different data sources online.

TABLE 2: RESULTS OF EXISTING METRIC WITH TEXTUAL APPROACH

Existing technique	Predicted Positive	Predicted Negative
Positive Cases	32	8
Negative Cases	4	6

Proposed technique	Predicted Positive	Predicted Negative
Positive Cases	34	6
Negative Cases	2	8

TABLE 3: RESULTS OF PROPOSED SUFFIX ARRAY TOKENIZATION APPROACH

Parameters	Metric with Textual Approach	Suffix Array Tokenization Approach
Precision	0.88	0.944
Recall	0.80	0.85
Accuracy	0.76	0.84

Using the more than four conditions, we have looked at the execution of our proposed approach and existing approach in view of table 2 and table 3. The got results are appearing in table 4. Figure 4, 5 and 6 illustrates the precision, recall and accuracy of the web based cloned files.

FIGURE 4: PRECISION RATIO OF PROPOSED AND EXISTING APPROACH

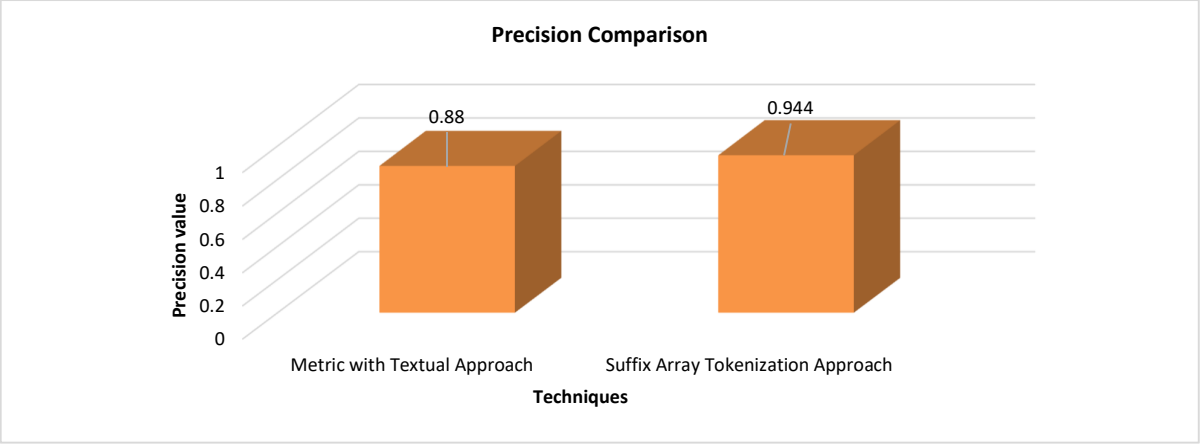


FIGURE 5: COMPARISON OF PROPOSED AND EXISTING APPROACH IN TERM OF RECALL

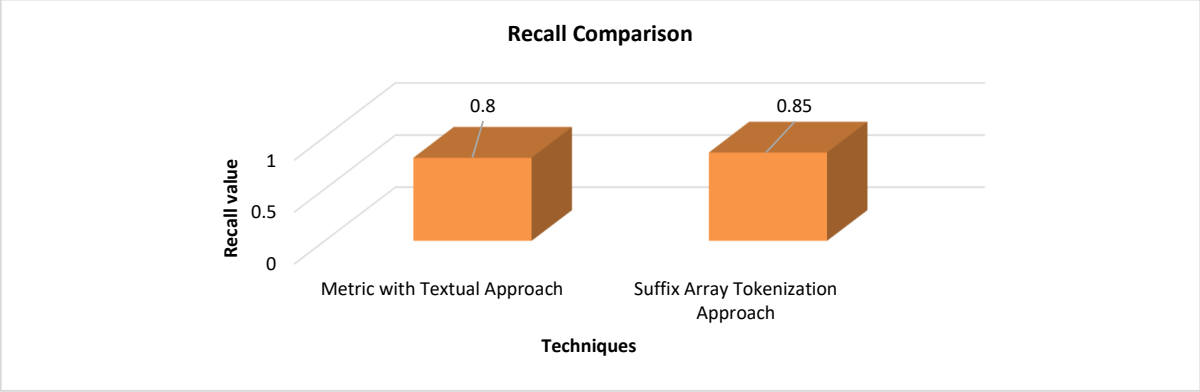
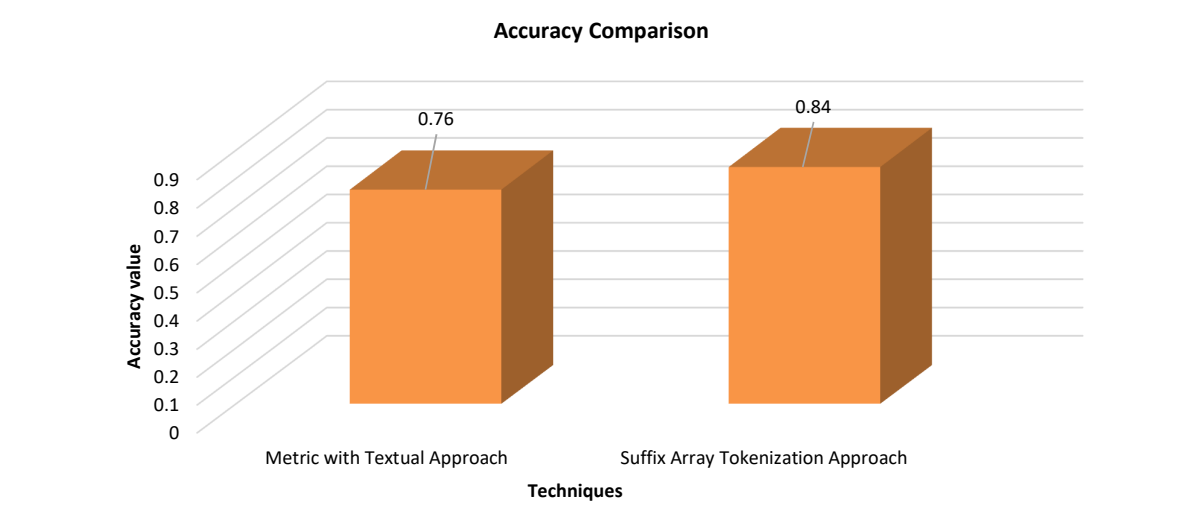


FIGURE 6: COMPARISON OF PROPOSED AND EXISTING APPROACH IN TERM OF ACCURACY



We have performed experiments on some gathered sample files to analyze the precision, recall and accuracy parameters to evaluate the performance of the proposed approach. We have observed that proposed tool is detecting more efficient number of clones as compared to existing technique, so cloning rate ascertained by our clone identifier tool is more.

CONCLUSION

Software clone is a phenomenon in large software system. It is usually caused by programmers copy and paste activities. The reason for the existence of clones in the source code is that making a copy of a code fragment is simpler and faster than writing it from scratch. In this paper, we have concentrated on clone detection systems and tools, giving a concise however extensive study and a speculative assessment in view of altering situations. Our technique can making sense of clones inside enormous supply codes and is unprecedented in its ability to hit upon code duplication free of the supply language. The tool Code Clone Detector created works just for the object oriented language code and it is anything but simple to utilize. In this introduced work metric based approach is utilized to distinguish the potential clones and after that upgrading the clone by means of utilizing token based technique.This approach is compared with source code of netbeans bellon and overshadows bellon individually in term of precision, recall and accuracy. From the simulation comes about it has been noted that the proposed approach give better outcomes as compared with existing methodology.

In future, this approach can be coordinated with different methodologies like abstract syntax tree based approach and the program dependence graph approach to deal with clone efficiently.

REFERENCES

1. Toshihiro Kamiya, Shinji Kusumoto, and KatsuroInoue (2000), "A Token-based Code Clone Detection Tool-ccfinder and its empirical evaluation", Technical report.
2. Cory Kapser, Michael W. Godfrey, "Cloning Considered Harmful (2006), "Considered Harmful", in proceedings of the 13th Working Conference on Reverse Engineering, pp. 19-28, Washington, DC, USA.
3. C.K. Roy and J.R. Cordy (September 2007), "A Survey on Software Clone Detection Research, "Queen's School of Computing", Technical Report No. 2007-541, vol. 115.
4. J.H. Johnson (1994), "Substring Matching for Clone Detection Tools, "In *Proceedings of the 10th International Conference on Software Maintenance"*, pp. 120– 126.
5. T. Kamiya, S. Kusumoto, and K. Inoue(July 2002), "CCFinder: A MultiLinguistic Token- Based Code Clone Detection System for Large Scale Source Code", IEEE Trans. Software Eng., vol. 28, no. 7, pp. 654-670.
6. B. Baker (2007), "Finding Clones with Dup: Analysis of an Experiment", IEEE Transactions on Software Engineering, vol. 33, no. 9, pp. 608–621.
7. Raghavan Komondoor and Susan Horwitz (July 2001), "Using Slicing to Identify Duplication in Source Code", in Proceedings of the 8th International Symposium on Static Analysis (SAS'01), Vol. LNCS 2126, pp. 40-56, Paris, France.
8. J. Mayrand, C. Leblanc and E. Merlo (November 1996), "Experiment on the Automatic Detection of Function Clones in a Software System Using Metrics", in Proceedings of the 12th International Conference on Software Maintenance (ICSM'96), pp. 244–253, Monterey, CA, USA.
9. C. K. Roy, J. R. Cordy, and R. Koschke (2009), "Comparison and Evaluation of Code Clone Detection Techniques and Tools : A Qualitative Approach", Science of Computer Programming, pp 470–495.
10. D. G. Devi, M.Punithavalli (2011) "Comparison and Evaluation on Metrics based Approach for Detecting Code Clone", Indian Journal of Computer Science and Engineering, Vol. 2 No. 5, pp. 747-751.
11. A. De Lucia, R. Francese, G. Scanniello, G.Tortora (2005), "Understanding Cloned Patterns in Web Applications", Proceedings of the 13th International Workshop on Program Comprehension (IWPC'05), IEEE.
12. G. Gupta and I. Singh, "A Novel Approach Towards Code Clone Detection and Redesigning (2013) "International Journal of Advanced Research in Computer Science and Software Engineering", vol. 3, no. 9, pp. 331–338.
13. T. Muhammad, M. F. Zibran, Y. Yamamoto, C. K. Roy (2013), "Near-Miss clone patterns in web applications: AN Empirical study with Industrial Systems", 26th IEEE Canadian Conference of Electrical and Computer Engineering (CCECE).
14. RubalaSivakumar, Kodhai. E (2012), "Code Clones Detection in Websites using Hybrid Approach", International Journal of Computer Applications, Vol. 48, No.13.
15. S. Ur. Rehman, R. khan,S. Fong and R. Biuk-Aghai (2012), " An Efficient New Multi-Language Clone Detection Approach from Large Source Code ",IEEE.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application & Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you to supply your critical comments and suggestions about the material published in this issue, as well as on the journal as a whole, on our e-mail **infoijrcm@gmail.com** for further improvements in the interest of research.

If you have any queries, please feel free to contact us on our e-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward to an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-
Co-ordinator

DISCLAIMER

The information and opinions presented in the Journal reflect the views of the authors and not of the Journal or its Editorial Board or the Publishers/Editors. Publication does not constitute endorsement by the journal. Neither the Journal nor its publishers/Editors/Editorial Board nor anyone else involved in creating, producing or delivering the journal or the materials contained therein, assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information provided in the journal, nor shall they be liable for any direct, indirect, incidental, special, consequential or punitive damages arising out of the use of information/material contained in the journal. The journal, neither its publishers/Editors/ Editorial Board, nor any other party involved in the preparation of material contained in the journal represents or warrants that the information contained herein is in every respect accurate or complete, and they are not responsible for any errors or omissions or for the results obtained from the use of such material. Readers are encouraged to confirm the information contained herein with other sources. The responsibility of the contents and the opinions expressed in this journal are exclusively of the author (s) concerned.

ABOUT THE JOURNAL

In this age of Commerce, Economics, Computer, I.T. & Management and cut throat competition, a group of intellectuals felt the need to have some platform, where young and budding managers and academicians could express their views and discuss the problems among their peers. This journal was conceived with this noble intention in view. This journal has been introduced to give an opportunity for expressing refined and innovative ideas in this field. It is our humble endeavour to provide a springboard to the upcoming specialists and give a chance to know about the latest in the sphere of research and knowledge. We have taken a small step and we hope that with the active co-operation of like-minded scholars, we shall be able to serve the society with our humble efforts.

Our Other Journals