



## An Effective Code Metrics for Evaluation of Protected Parameters in Database Applications

Varun K L Srivastava<sup>1</sup>, N. Chandra Sekhar Reddy<sup>2</sup>, Dr. Anubha Shrivastava<sup>3</sup>

<sup>1</sup>Director General, Department of R & D, Association of Education Training & Research Institute (ASSOED), Noida (UP) 201301, INDIA. [mr.varunsrivastava@gmail.com](mailto:mr.varunsrivastava@gmail.com)

<sup>2</sup>Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal Hyderabad - 500043. [nreddy28208@gmail.com](mailto:nreddy28208@gmail.com)

<sup>3</sup>Associate Professor, G. L. Bajaj Institute of Technology and Management, Noida (UP) 201301, India [anubha.shrivastava@glbitm.org](mailto:anubha.shrivastava@glbitm.org)

### ABSTRACT

Software protected may be the key part of product characterization similarly as the simplicity for which adjustments could be modified when the software will be conveyed. Track the protection performance of the software product is extremely difficult but it is generally accepted by the specialists. Huge study investigations predict that software which is object oriented can be achieved before original operation of software using design code proposed by Chidambaram and Kemerer (C&K). The suggested code metric suited may be evaluated to break down and observationally approved utilizing five software product systems. Those results indicate that those suggested metric suited may be thick, as viable experimental personal satisfaction mode to protected prediction for every one software system as a rule and for information escalated consideration product systems specifically. The suggested metric suited might be essentially supportive of the developers over examining the programming protected of information escalated consideration product frameworks in front of deploying them.

**Key words:** Code Metrics, Maintenance, Protected Index, Metrics evaluation, Empirical quality mode, Software protected.

### 1. INTRODUCTION

In an evolving domain, programming is additionally inclined to software protected to adjust programming support is single key procedures of the software life cycle. The explanation behind the product updates is to keep programming process, avert and revise flaws in the product and enhance the usefulness of the product. Support alludes to the alterations made to programming frameworks after their underlying discharge. It is unrealistic to build up a product framework that does not require support since change is the automatic nature of programming frameworks [1]. Protection is characterized by the IEEE as the

procedure of changing a product framework or part later conveyance to right blames, show signs of improvement execution or else different Characteristics, adjust to a modified domain. The protected is firmly identified. With programming support in light of the fact that the ease among which the safeguarding of the framework is perform hence protected. Present numerous endeavors to evaluate the protected of a product framework. This generally utilized programming metric which measures the practicality is recognized as protected index (MI).

The protected list might have been provided by oman and Hagemeister Also might have been settled on crazy from Different product measurements which may be the hotspot from claiming product protected. It comprises of a polynomial statement Furthermore brings about An number demonstrating the general schema protected khan and so forth to their book portrayed protected list Similarly as An gathering from claiming programming measurements to make particular McCabe's Cyclomatic many- agreed personal satisfaction (cc), Halstead's volume (v) Also lines from claiming code (LOC) that impact protection of the item. Protection list could alter starting with new code included of the present source book due to bugs adjusting alternately other medicinal exercises. Similarly as shown by Coleman an Protected list estimation for 85 reveals to that the result will be exceptionally maintainable, an estimation from claiming 85 Furthermore 65 moderate protected What's more An nature underneath 65 exhibits that the skeleton may be tough on keep up [2]. The protected list (MI) may be known as an estimation of protected. Run of the Plant estimations from claiming protected list ranges from 200 to-100s. Higher protected list is that reflection about better protected. The issue of foreseeing that protected from claiming modifying need was made once how protected might be foreseen by using separate instruments and methods toward the season about arranging by motivation behind programming outline measurements. Investigations build a connection among object-oriented

product measurements and its protection. They need additionally discovered that these measurements have a chance to be utilized Likewise interpreters about support exertion. Correct desire from claiming programming protected can be supportive because of the going with reasons.

**2. RESEARCH METHODOLOGY**

In this section, we illustrate those Different machine inclining (ML) systems utilized to settling on the prediction models and also on discover those relationship of plan measurements for protected. Late examination exercises conveyed by creators [7] [8] [9] [10] need uncovered that ANN will be extremely capable in classifying and more distinguishing those information patterns, so they need aid great suiting for prediction issues Similarly as On such instances Despite those needed learning will be was troublesome with point out Anyway sufficient information for perceptions would accessible to take in. They need aid initially produced with copy essential living neural frameworks especially the neurons exhibit in the human mind. Four diverse forms from claiming ANN models bring been chose in the present study similarly as said the following. (a) once more proliferation system (BPN): In spite of BPN may be initially imagined by Hu [13] in 1964 In any case it went under use just clinched alongside 1986 by Rumelhart et al. [11] when it might have been utilized Concerning illustration managed taking in techno babble. Preparing information to BPN comprises of couple of vector. Throughout those preparing course, an enter vector is introduced for those taking in methodology. Yield vector is created starting with these taking in and compared with that real focus vector. Though there is at whatever distinction in the principles, those weights of the organize need aid re-adjusted to decrease this slip and the methodology is repeater until the wanted yield will be prepared. (b) Kohonen organizes (KN): suggested by Kohonen [13], KN will be best known as self Sorting out networks similarly as they figure out to make maps of the enter space over an individual organized lifestyle. Although, KN will be developed will furnish an approach for speaking to multidimensional information Previously, significantly bring down dimensional spaces An metric suited for foreseeing programming Protected over information escalated consideration requisitions 9 system may be made that figure out the data such-and-such At whatever topological associations inside the preparation set are supported without supervision.

**3. THE GOAL OF THE STUDY**

As shown the table 1, those code dissection Metrix through object-based-C\$ code helps about code measurements to find possible exceptional protection throughout Different parameter such as Protected index, LOC, Cyclomatic Complexity, profundity from claiming

legacy, class combination and so forth throughout this way, observing and stock arrangement of all instrumentation . An fair-minded approach of this examine is will assess protected list for similarly object-oriented C# examinations What's more will figure out upgrading outcomes for those help for Visual studio code measurements. On assess consequences, create through code measurements and will upgrade protected, surveying code personal satisfaction previously, an object- turned modifying.

**Table 1:** The calculation of code metrics and Experiment

Projects count	Total Maintainability index	The Cyclomatic Complexity	In Depth of Inheritance	Coupling Class	Lines of Code
D1	14	13	2	2	36
D2	92	14	1	3	10
D3	90	15	1	3	13
D4	93	7	3	4	15
D5	89	9	1	3	16
D6	78	6	2	7	7
D7	69	8	2	1 2	17
D8	72	11	3	1 1	16
D9	56	26	4	1 2	34
D10	69	7	5	8	9
D11	80	9	6	8	7
D12	82	18	4	1 2	40
D13	83	17	2	2	24
D14	86	3	2	2	10
D15	75	5	1	1	11
D16	79	8	3	3	18
D17	69	5	2	2	13
D18	71	3	4	2	12
D19	72	6	1	2	15
D20	79	8	1	8	16
D21	86	14	1	4	18
D22	88	3	1	3	17
D23	81	1	1	2	12
D24	56	3	1	3	7
D25	57	4	1	2	21
D26	63	6	1	2	16
D27	65	3	1	1	8
D28	67	7	1	8	24
D29	78	4	1	1	17
D30	77	5	1	2	11
D31	75	6	1	1	18
D32	76	6	1	3	11
D33	77	7	1	4	12
D34	79	11	1	4	18
D35	81	12	1	4	19
D36	82	11	1	8	13

#### 4. RESEARCH METHODOLOGY

The source book for C# 40 analyses runs for Visual studio 2012. Uncover code measurements in different areas like protection guide, profundity of legacy, Cyclomatic Complexity and population combination and also Lines of code, Spellbinding ponder and relationship Research Methodology is used here. For Metrics, that is powerfully connected with other, for further consideration.

##### Descriptive Code metrix Methodology

A measure of variation (for dispersal) describes the spread or deviation of the entity values around the central value of a set of data. Therefore ,difference sets of data may have the same central value but differ greatly in terms of variation or dispersion.

Different measures of variation in the code metrix

There are following five different measures of variation:

- a) Range
- b) Quartile deviation (or semi- interquartile range)
- c) Significant divergence
- d) Normal divergence
- e) Variance

##### Correlation

Literally, correlation means an association of two or more facts. In statistics, the correlation may be defined as the tendency of simultaneous variation between two variables. The distribution involving two variables are called vicariate distribution and the distribution involving more than two variables are called multivariate distribution. In statistics, the degree of correlation between two or more variables is studied. Correlation can be defined according to the following points:

##### 4.1 Source Code Measurement

According to parameters, a result was concluded by code metrics. An arrangement of such cutoff points from various effective system composing rules characterizes a training standard, i.e., any correlation between these qualities can prompt an agent perspective of the nature of the experimented projects.

The metrics considered are among the comprehensive reports and utilized as a part of the information and are as following:

The Total Number of lines of code (LOC)

It events the physical size of the system code, barring clear lines and remarks.

##### 4.2 Cyclomatic Code Complexity V (g)

It was planned by McCabe, these metrics check the quantity of control flow graph of a project constituent. Cyclomatic complexity worth relies on the quantity of

twigs created by contingent proclamations (if-then-else). It activities have the structural complexity of the part.

##### 4.3 Protected Code guide (PCG)

MI events protected by focusing on the size, the intricacy, and the self-distinction of the code considered. A new protected index is obtained after experimentation of the current source code and a product, "new code" is formed. However, since MI depends on its features which are considered as qualities of software. It is similarly autonomous without a doubt; the span of these progressions might be used to analyze frameworks of disparate size. The coefficients of the process for Protected Index have been adjusted [12]. On different programming, frameworks kept up by Hewlett-Packard. Protected Index defenders confirmed this type of the MI perfect for the mathematical statement. In most cases, modern estimated delicate product frameworks are applied. High MI values show high Protected.

##### 4.4 Objects pairing

In 1974, Stevens et al. initially characterized combination in the setting of organized advancement as "the measure of the quality of affiliations built up by an interface particle from two distant modules". Combination is determined of association of two items. For instance, objects A and B are coupled if a strategy for questioning A calls a technique or gets to a variable in item B. Classes are attached when techniques report in one class reported strategies or characteristics of alternate classes.

##### 4.5 The Depth of Inheritance Tree (DIT)

The profundity of a class of the legacy progressive system is characterized as the most extreme length of the class hub to the root/guardian of the class order tree and is planned by progenitor classes. In cases including various legacy, the DIT is the most extreme length of the hub to the base of the tree [7].

#### 5. ANALYSIS, IMPLEMENTATION, AND VALIDATION

In this paper, show the objective to build object-oriented software as shown the table 2.

**Table 2:** Descriptive Analysis of Software Code Metrics

	Maintain Ability Index	Cyclomatic Complexity	Depth of Inheritance	Coupling Class	Lines of Code
Mean	75.18	7.45	1.12	4.40	15.38
Standard Error	2.11	0.86	0.07	0.54	1.32

Median	74.01	6.65	1.12	3.02	14.02
Mode	91.21	6.15	1.14	2.03	11.05
Standard Deviation	13.85	5.32	0.42	3.21	8.16
Sample Variance	194.72	27.56	0.15	11.23	66.85
Kurtosis	8.42	6.24	12.47	-0.05	2.66
Skewness	-2.12	2.12	3.46	1.05	1.58
Range	78.12	27.13	2.12	11.12	35.00
Minimum	15.35	2.14	1.14	1.00	6.00
Maximum	93.12	29.16	3.02	12.00	41.00
Sum	3047.20	309.17	45.30	180.00	615.00
Count	40.21	40.19	40.12	40.00	40.00
Confidence Level (95.0%)	4.46	1.68	0.13	1.08	2.61

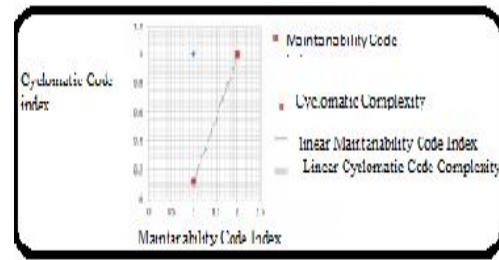
**Table 3:** Correlation between software code metrics

	Protected Index	Cyclomatic Complexity	Depth of Inheritance	Coupling Class	Lines of Code
Protected Code Index	1.02				
Cyclomatic Code Complexity	0.12	1.02			
In Depth of Inheritance	0.23	-0.04	1.00		
The Code Coupling Class	0.10	0.60	-0.12	1.00	
Lines of Code	-0.30	0.73	-0.08	0.28	1.00

As in table 3 & 4 Co-variation between two variables in opposite direction are negatively correlated with Depth of Inheritance and Cyclomatic Complexity are -0.03, Coupling Class and Depth in Inheritance are -0.12, Lines of code and Protected Index are -0.30, as well as Lines of codes and Depth in Inheritance, are -0.08.

**Table 4:** Correlation between protected index and cyclomatic complexity

	Protected Code Index	Cyclomatic Code Complexity
Protected Code Index	1	
Cyclomatic Code Complexity	0.114	1



**Figure 1:** Correlation between Protected Code Index and Cyclomatic Code Complexity

In the other way, the correlation between the two variables in which can be expressed by a straight line is called linear correlation as shown the figure 1.

### 5.1 Software Protected Metrics Help identify Problem Areas

Software protected needs more workout for the engineer and hence it belongs to the development of software lifecycle. A forty programming has done four sorts of maintenance on new arrangements or upgrades: remedial, versatile, perfective, and deterrent. These moves have made supplementary time to finish if the code is difficult to oversee in any case. A PC program with these capacities requires expanded programming: poor code quality, undetected vulnerabilities, source code imperfections, inordinate specialized multifaceted nature, vast frameworks, defective reported frameworks, over the top dead code. Asset needs to keep on developing as situations turn out to be continuously intricate and applications are immediately created. Software protected metrics give a modest, and exact way to deal with identifying conceivable reasons for un maintainable frameworks. Thus, researcher association gets vision into change regions and can screen basic frameworks in each part its application. The issue of foreseeing the protected of programming is largely recognized in the business and much has been composed on how protected can be evaluated by utilizing diverse devices. Procedures of the phases of planning with the assistance of software design metrics concentrates on how to manage and find out the firmness of connection between Object Oriented software metrics and its protected .They have moreover built up that these metrics can be appropriate as indicators of maintenance effort. Exact pection of software protected can be productive in view of the accompanying rationale

### 5.2 Discussion

This paper gives various updates for generalized utilization of Protected Index. To recap, keep on remarking the source codes, however, do not place an excess of confidence in remarks to enhance protected. Keep on measuring protected utilizing Protected Index without translating the outcomes in a vacuum. Know

about the confinements of target measurements, for example, MI. While changing advancements it requires evolving measurements. In designing, protected it should be kept in mind that deficiencies should overcome in future with regards that it benefit as much as possible from proficiency, dependability, and security, meet new necessities, make future upkeep less demanding, or deal with a changed situation. While designing it should be kept in mind that defective segments should be recovered in spite of substituting newer segments. Utilizing the MI to survey source code and recognizing and measure protected is a viable methodology. The protected index gives one little point of view of the profoundly complex issues of software maintenance. The MI gives a magnificent manual for direct human examination [6] [7] [8].

## 6. RESULT

The Protected Index of Experiments 40 was practical over statistical tests. The results show that two variables co-varying in the same direction are positively correlated i.e. a positive correlation between cyclomatic complexity and protected index, Depth of Inheritance and protected index, class coupling and protected index ,class coupling and cyclomatic complexity lines of codes and cyclomatic complexity and lines of codes and class coupling of code metrics. Similarly, Co-variation between the two variables in opposite direction is negatively correlated. The increase in one variable results in a corresponding decrease in the other .For example, increase in lines of codes results in a corresponding decrease in protected index, depth in inheritance and cyclomatic complexity ,class coupling and depth in inheritance as well as lines of codes in code metrics. Maximum the protected Index is 93 of experiment 2. Minimum Protected Index is 15 of experiment 1 after Statistical analysis of a set of 40 programs in c sharp object-oriented programming. Utilizing the Protected Index to evaluate a source code and consequently recognize and measure Protected is a compelling methodology. The Protected Index gives one little viewpoint of the exceptionally complex issues of Software maintenance.

## 7. CONCLUSION AND FUTURE SCOPE

Those objective about our Scrutinize might have been will observational inspect the adequacy from claiming new recommended metric suited to foreseeing product protected for information escalated consideration requisitions Likewise its significant to provide for rise to thoughtfulness regarding the database accesses with those expansion done information and in addition those number from claiming times information get accessed. The Protected Index gives a brilliant manual for direct human examination. This paper gives various considerate to the pragmatic utilization of Protected Index.

Throughout attempting in this zone about investigation, a considerable measure about extent for future fill in need

been watched. There is requirement of further experimental investigations will Figure protected list. One technique for testing this objective is by controlled test and investigation. Programming designing endeavors the cost, build dependability, and expansion heartiness and in addition to other things .The objective of this analysis is to grow the establishments of software designing so those work with programming can settle on savvy decisions when fabricating and keep up Systems. Similar Experiments can be carried out for protected index.

## REFERENCES

1. Malhotra R. and Chug A.(2014), "A Metric Suite for Predicting Software Protected in Data Intensive Applications", Kim H.K.,AO S., Amouzegar M.A.(eds.), *Transactions on Engineering Technologies* , Springer Netherlands, pp. 161-175. [https://doi.org/10.1007/978-94-017-9115-1\\_13](https://doi.org/10.1007/978-94-017-9115-1_13)
2. Kumar L. and Rath S.K.(2015), "Neuro-Genetic Approach for Predicting Protected Using Chidamber and Kemerer Software Metrics Suite" , Herwig U., Phayung M., Sirapat B., (eds.),*In Recent Advances in Information and Communication Technology*, Springer International Publishing, 361,pp. 31-40 [https://doi.org/10.1007/978-3-319-19024-2\\_4](https://doi.org/10.1007/978-3-319-19024-2_4)
3. Coleman D., Ash D., Lowther B. and Oman P. (1994), "Using metrics to evaluate software system protected, "Computer", 27(8), pp.44-49. <https://doi.org/10.1109/2.303623>
4. Welker K.D., (2001), "The software protected index revisited", *CrossTalk* , 14, pp.18-21.
5. Ganpati A., Kalia A. and Singh H.(2012), "A Comparative Study of Protected Index of Open Source Software", *Int. J. Emerg. Technol. Adv. Eng.*, 2(10), pp.228-230
6. Aggarwal KK, Singh Y, Kaur A, Malhotra R. Application of Artificial Neural Network for Predicting Protected Using Object Oriented Metrics. Proceedings of World Academy of Science, Engineering and Technology 2006, 15: 285- 289
7. Malhotra R, Chug A, Software Protected Prediction using Machine Learning Algorithms, *Software Engineering: An International Journal*, 2(2), (2012), pp 19-36.
8. Sun P, Wang A, Application of Ant Colony Optimization in Preventive Software Maintenance Policy, 2012 IEEE <https://doi.org/10.1109/ICIST.2012.6221624>
9. Malhotra R and Chug A, An Empirical Study to Redefine the Relationship between Software Design Metrics and Protected in High Data Intensive Applications, *Lecture Notes in Engineering and Computer Science : Proceedings of The World Congress on Engineering and Computer Science 2013, WCECS 2013, 23-25 October, 2013, San*

Francisco, USA, pp. 61-66

10. Rumelhart DE, Hinton GE, Williams RJ, Learning internal Presentation by back propagating errors, The PDP research Group, Parallel Distributing Processing: Exploration in the Microstructure of cognition, MIT Press, MA, 1994
11. James Carbaugh<sup>1</sup>, Matthew Fletcher<sup>1</sup>, Raluca Gera,” Extracting Information Based on Partial or Complete Network Data”, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.1.1, 2019.
12. Henry S., Humphrey M. and Lewis J. (1990), “Evaluation of the protected of object-oriented software”, Liu Y.W.(eds.),10<sup>th</sup> Conference on *Computer and Communication Systems, TENCON'90*, Hong Kong, Sept 24-27, 1990. IEEE, pp. 404-409.
13. Kohonen T , Self-Organization and Associative Memory, Berlin: Springer, 1989  
<https://doi.org/10.1007/978-3-642-88163-3>