International Journal of Engineering & Technology, 7 (2.30) (2018) 1-5

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Empirical model for quantification of confidentiality in OO system

Rakesh Kumar [1] *, Dr. Hardeep Singh [2]

[1] Assistant Professor, Dept. of Computer Science, Khalsa College for Women, Amritsar, India PhD Scholar Socis, Ignou
[2] Professor, Dept. of Computer Science & Engineering Guru Nanak Dev University, Amritsar, India
**Corresponding author E-mail: rakeshmaster1980@rediffmail.com*

Abstract

The coupling or aggregation binds together the different entities or components within the system. An external process when takes or try to take the control of the system will be assisted in its action if the underlying system is highly coupled. A highly coupled design degrades the ability of software to defend against exploitation. Thus from a software developer's point of view, we must provide so much security at design time that no one outside the system should be able to access in unauthorized way. It is to insure that information leakage is minimal (if not zero as is desired theoretically). This research work done quantitatively, describes the ability of object oriented coupling metrics to predict faulty classes. There are two major section of this paper. One section covers the ability of multi layer neuron perceptron model for prediction of faulty classes and in other section we have proposed and validated a statistical model for confidentiality using data set of dif-ferent releases of apache velocity project so as to quantify the effects of coupling on confidentiality of system.

Keywords: *Bugs; Confidentiality; Coupling; Metrics; Software Security*.

1. Introduction

The software vulnerabilities are prominent reasons behind the software failures and poses risk to its security. The timely and accurate defect prediction helps a developer to incorporate security within the application Software. The sloppy coding or integration with other modules provides opportunities to malevolent hackers to exploit these security breaches [11]. The non modular and tightly coupled systems are more vulnerable to errors. They need more efforts to maintain and it is hard to augment them [1] [3].

The coupling is an important traditional internal software property and is linked with security which is an external software property [4]. The software architect's should focus on coupling property when security is elicited as an important factor of software architecture. It has been observed that highly coupled design degrades the ability of software to defend against exploitation [5].

2. Literature review

Ivan & Krsul [16] defined security vulnerability as an instance of fault in specification, configuration or development of software. Execution of this type of instance can results in implicit or explicit security policy violation. These flaws can ripple through the design if coupling is extensively used in the development that could be exploited or attacked. Author showed the abilities of statistical analysis and machine learning tools to discover patterns and regularities about the nature of vulnerabilities. Their study divides vulnerabilities into four hierarchical classes, design flaws, environmental flaws, coding flaws, and configuration flaws.

The "attack ability" as observed by Liu and Traore [20] is extent to which software could be target of successful attacks. It is an indicator of vulnerability. An early removal of vulnerability is an indicator of secure design. There is evidence that shows internal attribute such as coupling may be source of vulnerability and attack ability. Authors used regression analysis to establish that there is a strong correlation between attack ability and coupling.

It has been explored by Ayanam [6] that many forms of SQL injections, Denial of service attacks and Buffer overflow attacks are caused by exploitation of certain type of coupling. The AV and DV metrics has been proposed and validated with empirical data as an aid to act as indicator of security vulnerability. The usage of internal software attribute, coupling, to predict external attribute, security, is well elaborated in this study. The paper observes coupling as an important attribute while designing software with high security as a requirement.

Chowdhury and Zulkermine [10] conducted an experiment to prove their hypothesis that the coupling metrics are positively correlated to security vulnerabilities in software. They used spearman rank correlation between coupling metrics and vulnerabilities data for Mozilla Firefox. Their empirical study involved coupling, cohesion and complexity metrics and machine learning techniques for prediction of security level for the software system. It was observed that highly coupled files have higher number of vulnerabilities. The correlation values obtained were ranging from 0.434 to 0.539.

Lagerstrom et al. [18] said that the usage of size and complexity metrics for defect prediction has proved to be quite effective. There is a scarcity of work that relates vulnerabilities with software architecture. This study explored the vulnerabilities using code churn, cyclomatic complexity and coupling metrics such as direct, indirect and cyclic coupling. This empirical study has used Google chromium project's data set. A strong relation has been observed between architecture coupling metrics and vulnerabilities. These software vulnerabilities when exploited, can lead to loss in confidentiality, integrity and availability parameters. The variable

Copyright © 2018 Rakesh Kumar, Dr. Hardeep Singh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

such as size, complexity, code churn and coupling are related with vulnerabilities and thus likely to cause the increment in vulnerability incidents.

Sullivan & Chillarege [23] said that the recent times has seen a lot of customer outage due to software defects and mismatch in pace with hardware quality and reliability. There is a lack of clear methodology to recover from and avoid software defects. The data set used is a high end operating system product. This study explored that the impact of overlay defects is higher than that of a regular defects. These defects cause the software failures. The boundary condition and allocation management are observed as primary reason of overlay defects.

Wilkie & Kitchenham [25] described an empirical analysis of class coupling on changes made to a C++ application over a time span of two and half year. The CK metric, CBO, is used here to measure class coupling within the application . An in depth study of ripple effect of change to source code and coupling is provided. It is observed that CBO measure is able to identify the most change prone classes. The metric RFC provides an assessment of coupling between member functions within a class.

Thapaliyal & Verma [24] elaborated that OO metrics are useful to analyze and predict about quality of software. The object oriented metrics treats functions and data as combined object. This study evaluated two metrics namely CBO and WMC from CK metrics suites for their ability to predict defects. For data sets, 50 Java classes of different projects were taken. Study explored class size as a significant measure having positive impact on defects. The variables CBO and WMC are having insignificant relations, which is not consistent with earlier findings. It is observed that even the most recognized metrics may not be significant enough with diverse types of projects.

Jureczko [15] analysed significance of different process and product metrics in defect prediction modeling. They used 15 open source and 7 proprietary projects for their empirical analysis. The different metrics and defects were used to evaluate the Pearson correlation coefficients. The product metrics such as LOC, RFC, CBO, CAM and AMC were found to be very useful in defect prediction. Also a number of studies have used process metrics such as NR, NDC, NML and NDPV. The process metrics are correlated with defects on 0.3 or higher level of median. The metrics NDC and NDPV are found to have greatest discriminant power.

Agrawal and Khan [2] recommended that the security related decision should be taken at design phase of development. There is a lack of work to assess the effects of OO design characteristics on security, though the effort has already been made for quality. This study emphasizes about significance of coupling metrics and their role in inducing vulnerability propagation in OO design. The authors focus on minimizing the vulnerability of an OO design by controlling their propagation via coupling. CIVPF is assessed so that unwanted coupling can be avoided to come up with a better design.

Lori MacVittie [21] explored the use of loose coupling to legacy web applications and insisted that loose coupling can improve the security of legacy applications. The decoupling of security policies from other services preserves the ability to reuse services in multiple applications. Highly coupled integrated security policies can hinder availability of existing applications making use of that service when they are tested and redeployed. Loose coupling also provides configuration and maintenance benefits by decoupling many security functions from applications. The decoupling of security policy from applications decreases the potential for introducing new flaws when modifying the security code. The findings can be summed up by stating that loose coupling implementation helps bind security tightly to your applications.

M Alenezi and Ibrahim Abunadi [3] worked on problem of application security. The development of secure code was emphasized. They developed an approach to predict vulnerable code so as to prioritize the security efforts in development. They performed empirical analysis through WEKA tool using data set of three open source PHP web application projects. Their result shows that metrics can discriminate between and are predictive of vulnerabilities and can suggest for improvement of code and development team.

The techniques such as logistic regression, naïve bayes approach, random forests, KNN, radial basis function, multilayer perceptron [17], BBN and univariate logistic regression has been selectively used for developing the prediction models [14][26].

3. Collection of data

Data set is collected from three versions of Velocity software project so as to examine the ability of coupling metrics to predict the fault proneness of classes. The data set for velocity 1.4, Velocity 1.5 and Velocity 1.6 application has been used for this research work.

4. Research methodology

In this section, we will design a feed forward NN models using above mentioned data set of section 3. The set of metrics rfc, ce, IC, dam, cbm will be considered as the independent variables whereas defects as dependent variable to develop the required model. The proposed model will help to discriminate the defective module in software. Later on the model will be evaluated with different versions. We will use the calculated values of area under curve and accuracy of model for measuring the effectiveness of model.

5. Neural network model based upon coupling metrics

We will use Feed Forward Neural Network for predicting the defects. A FFNN consists of multiple layers of neurons. The model works on back-propagation learning algorithm. The connection between the ith and jth neuron is denoted as wij; the weight coefficient emphasizing the level of importance between ith and jth neurons. The output of a layer is determined by following equation:

a = x1w1+x2w2+x3w3... +xnwn

The “tansig” activation function is used here for hidden layer and the “purelin” activation function for output layer is preferred.

5.1. Experimental validations of effects of coupling metrics to identify faulty classes using neural network

As per discussion in section 4, our FFNN model is depicted in figure 1.

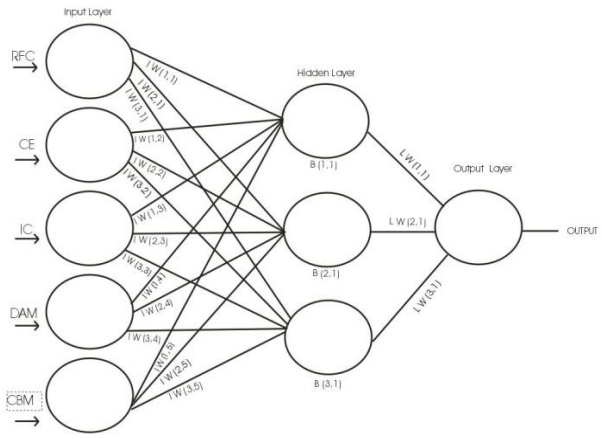


Fig. 1: Neural Network Model Based Upon Coupling Metrics.

In FFNN the neurons in a given layer are connected to all other neurons in subsequent next layer. The metrics used for the model are rfc, ce, IC, dam, cbm.

5.2. Results of implementation

The table 1 represents the value of confusion matrix obtained after completion of 1000 and 2000 epoch of experiment while testing on data set of velocity 1.4 projects.

Table 1: Confusion Matrix at 1000 and 2000 Epoch for Velocity 1.4 Data Set

	1000 epoch		2000 epoch	
	NF	F	NF	F
Nf	22	27	18	31
F	7	140	1	146

The MSE values of 0.74711 and 0.73357 at 1000 and 2000 epoch were observed in calculation which shows improvement in results with increase in number of iterations.
The area under curve values for above mentioned data sets are 0.567 and 0.521 respectively (see figure 2).

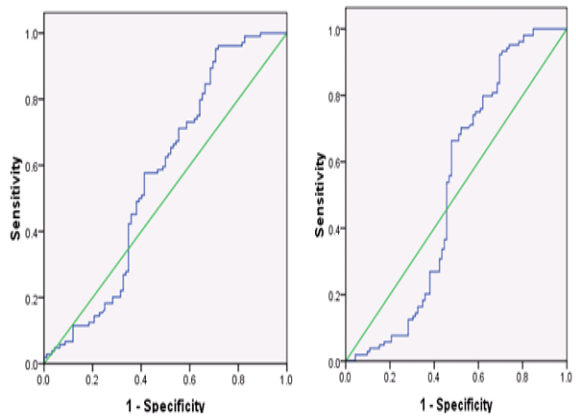


Fig. ROC curve for coupling at 1000 epoch for Velocity 1.4 with AUC is 0.567.

Fig. ROC at 2000 epoch for coupling metrics for velocity 1.4 with AUC is 0.521.

Fig. 2: ROC Curve and AUC Plots for Velocity 1.4.

This model was then validated on data set of velocity 1.5 application to produce confusion matrix table 2 at 1000 and 2000 iterations respectively.

Table 2: Confusion Matrix at 1000 and 2000 Epoch for Velocity 1.5 Data Set

	1000 epoch		2000 epoch	
	Nf	F	Nf	F
Nf	0	72	0	72
F	0	142	1	141

The AUC values for velocity 1.5 applications are compared as in figure 3.

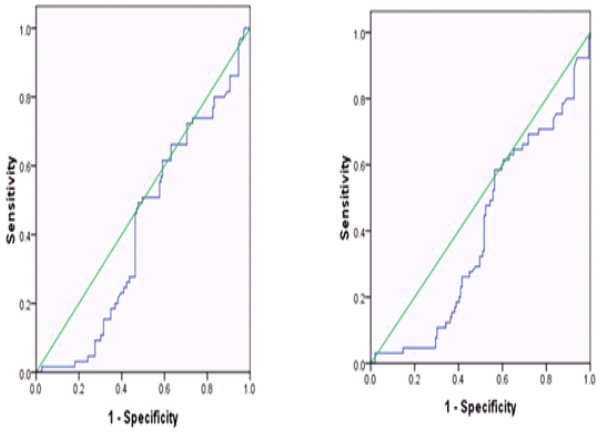


Fig: ROC curve for velocity 1.5 for coupling data at 1000 epoch and with AUC is 0.421

ROC curve for velocity 1.5 for coupling data at 2000 epoch with AUC is 0.397

Fig. 3: ROC Curve and AUC Plots for Velocity 1.5.

The same method was repeated to get results for data set of velocity 1.6 projects. The analysis of Coupling based model using data set of velocity 1.6 produces following values for confusion matrix.

Table 3: Confusion Matrix at 1000 and 2000 Epoch for Velocity 1.6 Data Set

	1000 epoch		2000 epoch	
	NF	F	NF	F
NF	75	76	129	22
F	10	68	44	34

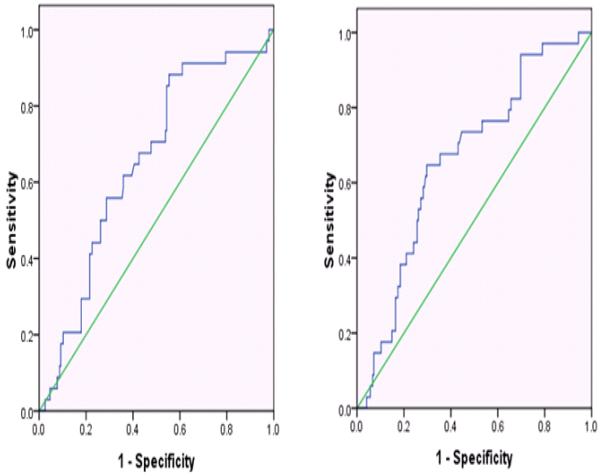


Fig: ROC curve for velocity 1.6 for coupling data at 1000 epoch with AUC is 0.650

Fig: ROC curve for velocity 1.6 for coupling data at 2000 epoch with AUC is 0.660

Fig. 4: ROC Curve and AUC Plots for Velocity 1.6.

Our aim was to study the effectiveness of coupling metrics on basis of NN model to identify the fault prone classes in the various versions' data. The model developed for one version was applied on other version's data too. We can deduce from results of the predicted model that the model is more accurate for version 1.4 and 1.6 than version 1.5. The mean square error analysis is as in table4 below.

Table 4: Mean Square Errors Values

Data Set	1000 epoch	2000 epoch
Velocity 1.4	0.74711	0.73357
Velocity 1.5	2.2578	2.2807
Velocity 1.6	1.9761	1.8559

6. Model for quantification of effects of coupling metrics on confidentiality of system

Confidentiality is among one of the major factor affecting the security of software. It is defined by experts as the unauthorized disclosure of information. So from software developer's point of view, we must provide so much security at design time that no one outside the system should be able to access in an unauthorized way. It is to insure that information leakage is minimal if not zero (which is desired theoretically). The coupling or aggregation binds together the different entities or components within the system. An external process when takes or try to take the control of the system will be assisted in its action if the underlying system is very highly coupled. This intruder process will be assisted by concerned entities and supporting services [19] [22]. The measurements about the various entities and their behaviour can be obtained through design constructs such as coupling, cohesion, encapsulation and polymorphism [12] [13]. The software security team through this model based on coupling metrics can quantify the confidentiality for the system under development so as to improve its overall security [7] [8] [9]. The earlier work done on coupling metrics has associated it with software maintenance. It is observed that highly coupled system is difficult to maintain. Its high value also hinders the developers from reusing the software. In other words the confidentiality is at risk of exposure in next version of the software too if not removed to significant level in system under development.

We have used coupling metrics namely cbm, rfc, ca, dam, ce, IC, cbo as independent variables (predictors) .The dependent variable is bug. The model is developed on basis of metric data set for application Velocity 1.4 obtained through PROMISE data repository.

The basic assumption before applying the multi linear regression is that the data shouldn't be collinear. A collinear data means concerned variables are also related to each other or are similar.

The VIF analysis and tolerance are widely recommended measure of the degree of collinearity. These variables which carry higher such values are referred as dummy variables. The VIF analysis quantifies the degree or severity (see table 5) of multi collinearity in an ordinary least square regression analysis. It measure how much the variance of an estimated regression coefficient increases due to collinearity.

Table 5: VIF Benchmark Values

Sr No	VIF value	Status of predictors
1	VIF=1	Not correlated
2	1<VIF<5	Moderately correlated
3	VIF>5 to 10	Highly correlated

The tolerance and VIF analysis on the selected metrics produced following table labelled as table 6.

Table 6: Collinearity Analysis for Coupling Metrics

Metric	Collinearity Statistics	
	Tolerance	VIF
Cbo	0.053	18.712
Rfc	0.540	1.853
Ca	0.077	12.979
Ce	0.174	5.753
IC	0.160	6.245
Dam	0.810	1.234
Cbm	0.162	6.155

In Table 6, the values 18.712 and 12.979 are not following the prescribed range for VIF values and also the Tolerance statistics are not significant for the variables cbo and ca. After eliminating the variables whose value of VIF is outside the range (one <VIF< 10), we get following result:

Table 7: VIF Analysis with Five Set of Coupling Metrics

Metric	Collinearity Statistics	
	Tolerance	VIF
Rfc	0.553	1.809
Ce	0.579	1.727
IC	0.163	6.145
Dam	0.860	1.163
Cbm	0.163	6.146

6.1. The proposed regression model to quantify confidentiality

We have applied multiple regression technique for finding the weight values for coupling metrics namely rfc, ce, IC, dam, and cbm. The un-standardized coefficients including constant value and standard error are tabulated as below using SPSS 20.0.

Table 8: The Regression Model for Rfc,Ce, IC, Dam, Cbm.

Model	Unstandardized Coefficients	
	B	Std. Error
(Constant)	1.065	0.102
Rfc	-0.006	0.003
Ce	0.030	0.012
IC	-0.451	0.185
Dam	0.351	0.141
Cbm	0.036	0.125

From the table it is clear that constant value is 1.065 and the coefficients for rfc is -0.006, for ce is .030, for IC is -0.451, for dam 0.351 and for cbm is 0.036. The model thus obtained by multi linear regression can be written as following:-

$$\text{Confidentiality} = 1.065 - 0.006 * \text{rfc} + 0.030*\text{Ce} - 0.451*\text{IC} + 0.351*\text{dam} + 0.036*\text{cbm} \quad (1)$$

The un-standardized coefficients are computed using SPSS and rfc, Ce, Ic, dam and cbm are independent variables chosen for the model by mathematical analysis as explained above in VIF analysis section. Now we have to apply the model to metric data of Velocity 1.4. After application of confidentiality model (1) on all the 196 classes of Velocity 1.4 and 214 classes of Velocity 1.5 data, we apply z-test,(In SPSS the Z-test analysis is available under the name of t-test, Z-test is applied when sample size is more than 30,otherwise t-test is preferred).It is required to check for normality of data in order to proceed for Z-test. We, for sake of test of normality of data, combine data for confidentiality for velocity 1.4 and 1.5 under one column in SPSS(named here as Confi14 column) .The results as shown in following figure clearly signify that the data is normal and hence appropriate for applicability of t-test for large sample(n>30).

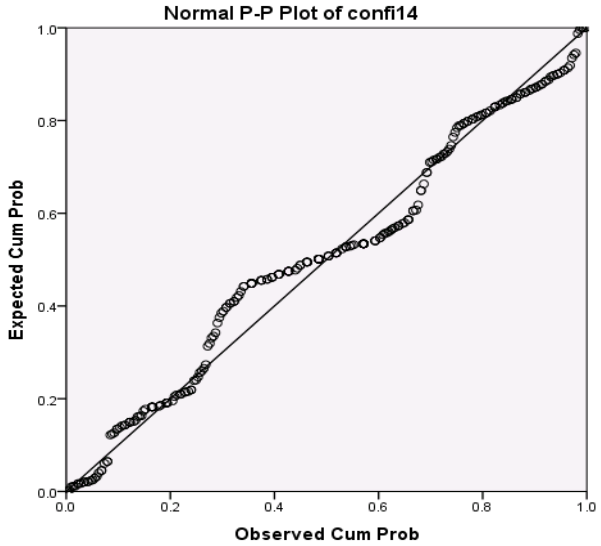


Fig. 5: P-P Plot for Test of Normality of Data.

6.2. Validation of confidentiality model

The Levene's Test for Equality of Variances shows that F value is 1.799 with significance 0.181. The t-test statistics obtained shows that the p value is equal to 0.163; it is clearly greater than 0.05, so there is no significant difference among data for Confidentiality values for velocity 1.4 and velocity 1.5 applications. The acceptance of model signify the validation of model .So model is highly acceptable for calculating the confidentiality of system based upon object oriented construct of coupling metrics.

7. Conclusion

Theoretical studies have shown that a low value of coupling metrics is preferable. Our study says that a value close to 0.8077 or less for the Confidentiality model indicates that combine effect of coupling metrics is producing less defects in the system. The higher values for metrics Ce, dam and cbm means the design is not secure. It is observed that, combined effect of the metrics is more influential than their individual effects on confidentiality of system based upon object oriented construct of coupling metrics. The model works well for within the company projects and more work is expected in the field to develop similar model for cross company projects.

References

[1] Abreu, F. B., Pereira, G., & Sousa, P. (2000). A Coupling-Guided Cluster Analysis Approach to Reengineer the Modularity of Object-Oriented Systems. *Proceedings of conference on Software Maintenance and Reengineering (CSMR'00), (pp. 13-22).* Zurich, Switzerland.

[2] Agrawal,A., & Khan, R.A. (2012). Role of Coupling in Vulnerability Propagation-Object Oriented Design Perspective. *Software Engineering: An International Journal (SEIJ), 2*(1), 60-68.

[3] Alenezi, M. & Abunadi, I. (2015). Evaluating software metrics as predictors of software vulnerabilities. *International Journal of Security and Its Applications*, *9*(10), 231–240.

[4] Allen, E. B., Khoshgoftaar, T. M., & Chen, Y. (2001).Measuring coupling and cohesion of software modules: an information-theory approach. *Proceedings of seventh International Software Metrics Symposium (METRICS'01), (pp. 124-134).*

[5] Arisholm, E., Briand, L. C., & Foyen, A. (2004). Dynamic coupling measurement for object-oriented software. *IEEE Transactions on Software Engineering, 30*(8), pp. 491-506.

[6] Ayanam, V. S. (2009). Software Security Vulnerability vs Software Coupling: A Study with Empirical Evidence. Master's Thesis, Southern Polytechnic State University, Marietta, Georgia, USA.

[7] Briand, L., Wust, J., & Louinis, H. (1999). Using Coupling Measurement for Impact Analysis in Object-Oriented Systems. *Proceedings of IEEE International Conf. on Software Maintenance, (pp. 475-482).*

[8] Briand, L.C., Daly, J., Porter, V., & Wust, J. (1998). Predicting fault-prone classes with design measures in object-oriented systems. *Proceedings of the Ninth International Symposium on Software Reliability Engineering (Cat. No.98TB100257), (pp.334-343). Paderborn.* doi:10.1109/ISSRE.1998.730898

[9] Cartwright, M., & Shepperd, M. (2000). An empirical investigation of an object-oriented software system. *IEEE Transactions on Software Engineering, 26 (*8), 786-796. doi: 10.1109/32.879814

[10] Chowdhury, I., & Zulkermine, M. (2011) .Using Complexity, Coupling and Cohesion metrics as Early Indicators of vulnerabilities. *Journal of Systems Architecture, 57*, 294-313.

[11] Devanbu, P.T., & Stubblebine, S. (2000). Software engineering for security: A roadmap. *Proceedings of the Conference on the Future of Software Engineering* (ICSE '00) (pp. 227-239). NY, USA: ACM. doi=http://dx.doi.org/10.1145/336512.336559

[12] Emam, K. El., Benlarbi, S., Goel, N., Melo, W., Lounis, H., & Rai, S.N. (2002). The optimal class size for object-oriented software. *IEEE Transactions on Software Engineering, 28(*5*),* 494-509. doi: 10.1109/TSE.2002.1000452.

[13] Evanco, W. M. (2003). Comments on The confounding effect of class size on the validity of object-oriented metrics. *IEEE Transactions on Software Engineering*, *29*(7), 670-672. doi: 10.1109/TSE.2003.1214331.

[14] Fenton, N.E., & Neil, M. (1999). A critique of software defect prediction models. *IEEE Transactions on Software Engineering, 25(*5*),* 675-689. doi: 10.1109/32.815326

[15] Jureczko, M. (2011). Significance of different software metrics in defect prediction. *Software Engineering: An International Journal, 1*, 86-95.

[16] Krsul, I. V.(1998).Software Vulnerability Analysis, PhD Thesis, Purdue University, West Lafayette, Indiana, USA.

[17] Kumar, V., Sharma, A., & Kumar, R. (2013). Applying soft computing approaches to predict defect density in software product releases: An empirical study. *Computing and Informatics, 32*, 203–224.

[18] Lagerström R., Baldwin C., MacCormack A., Sturtevant D., & Doolan L. (2017). Exploring the Relationship between Architecture Coupling and Software Vulnerabilities. *In: Bodden E., Payer M., Athanasopoulos E. (eds) Engineering Secure Software and Systems. ESSoS 2017. Lecture Notes in Computer Science, vol 10379*.Springer.

[19] Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *IEEE Transaction on Software Engineering, 34(*4*),* 485-496.

[20] Liu, M. Y. & Traore, I. (2006).Empirical Relation between Coupling and Attackability in Software Systems: A Case Study on DOS. *Proceedings of 2006 Workshop on Programming Languages and analysis for Security, (pp. 57-64). Ottawa,* Canada.

[21] Macvittie,L. (2008, March 18). Application Security: Loose Coupling for Legacy Apps [Blog Post]. Retrieved from https://devcentral.f5.com/articles/application-security-loose-coupling-for-legacy-apps

[22] Olague, H.M., Etzkorn, L. H., Gholston, S., & Quattlebaum, S. (2007). Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes. *IEEE Transactions on Software Engineering*, *33*, 402-419. doi: 10.1109/TSE.2007.1015

[23] Sullivan, M., & Chillarege, R. (2000). Software Defects and Their Impact on System Availability: A Study of Field Failures in Operating Systems. Digest of Papers - FTCS (Fault-Tolerant Computing Symposium).

[24] Thapaliyal, M. & Verma, G. (2010). Software Defects and Object Oriented Metrics:An Empirical Analysis. *International Journal of Computer Applications, 9*(5).

[25] Wilkie, F.G., & Kitchenham, B.A. (2000). Coupling measures and change ripples in C++ application software. *Journal of Systems and Software*, *52*(2-3), 157-164, https://doi.org/10.1016/S0164-1212(99)00142-9.

[26] Zimmermann, T., Nagappan, N., Gall, H., Giger, E., & Murphy, B. (2009). Cross-project defect prediction: a large-scale experiment on data vs. domain vs. process. *Proceedings of ESEC/ FSE, (pp. 91-100).* New York: ACM. http://dx.doi.org/10.1145/1595696.1595713.