Web-of-Service Software Reusability Prediction using Heterogenous Ensemble Classifier

Prakash V. Parande, M. K. Banga

Abstract: ***WoS computing environment is expected to have numerous parallel computing engines. Presently, software professionals or developers often want to reuse existing software components to exhibit a task with time-efficient and cost effective solutions. However, software component reusability in uncontrolled manner leads to failure, pre-mature shutdown and software smells or aging. This paper develops a novel evolutionary computing assisted ensemble classification system for WoS software reusability prediction. This applies different base learners such asNaïve Bayes (NB), Linear Regression (LR), Decision Tress (DT),Logarithmic Regression (LOGR),and Support Vector Machine (SVM),Multivariate Adaptive Regression Spline (MARS). Once training the base learners, the outputs of each classifier have been processed with majority vote.The computation in conjunction with weighted sum enabled final labelling of each software class. The performance results affirmed that the present work ensemble classifier has better performance with respect to base classifiers.***

Keywords: ***Ensemble classifier; heterogeneous ensemble; web-of-service; software reusability prediction.***

.

1. INTRODUCTION

The vital roles and high ability makes software an inevitable entry in modern scenario. The software component signifies anon-dependent section of software application in the form of set of codesand standardized parts, classes, and tests cases etc [1]. The process of reusing existing functionsto achieve efficient software solution for computational time and resource cost is called software reusability [2]. However, the excessive or uncontrolled reuse of software component could lead faults or aging related symptoms, sometimes called software smells [2][3]. In such cases assessing reusability extant of each software component can help avoiding aging or fault like events [4]. The work considers Object Oriented Programming (OOP) concept with software metrics such asCoupling between Object (CBO), Depth of Inheritance Tree (DIT), and Weighted Method per Class (WMC), Line of Code (LOC), Number of Children (NOC), and Lack of Cohesion in Methods (LCOM) [5-8]. In ensemble of classifiers,single decisions are merged to categorize new examples which are the base/weak classifiers [9][10]. Ensembles are familiar to lower the risk of opting the unethical model by clustering all candidates [9][10][13].

In feature-level fusion, attributes are entangling from multiple event info or data sets, and merged into anindividualappended feature vector[11]. Brooks et.al., [12] described feature-level fusion as ahighoptionfor correlated data and DLF is uncorrelated thusit's a better option. Performing training of each base learner over data subsets, the ensemble has been prepared using majority vote and weighted sum method. Performing two-class classification each class has been predicted as REUABLE and NON-REUSABLE. The overall proposed system has been developed using MATLAB 2017a software where the performance analysis affirmed that the proposed ensemble concept exhibits better performance than the classical base learners.

2. RELATED WORK

Singhani et al. [15] examined various factors influencing inspection of OOP software for an Analytical Hierarchical Process (AHP) model. OO-CK metrics can achieve software re-usability estimation and quality characterization [16-18]. Definition of threshold values for each CK metrics can enable fault resilient reusability prediction [19, 20]. In [21, 22] CK metrics is conceptualized to identify minimal OOP parameters of software design and reusability was predicted in each class. Zahara et al. [23] used Machine learning regression algorithms to perform reusability prediction for independent variables characterizing reuse-proneness of the classes in OOP software metrics. Torkamani [24] explored the possibility of other OOP-software metrics to perform reusability prediction. Similarly, it is also recommended OOP metrics for suitable software design[25]. There is a derived metrics of software to examine quality features of software parts in customizability, reusabilityand complexity[26]. They applied interface mechanism to estimate Component Reusability (CR) and Component Ruse Level (CRL) to compute the reuse proneness level. Line of Code (LOC) metric as the

Revised Manuscript Received on July 25, 2019.

Prakash v. Parande,
Research Scholar, School of Computing & Information Technology, Reva University, Bangalore-560064, India, prakashvp2010@gmail.com
Dr. M. K. Banga,
Ex - Director of School of Computing and information technology, Reva University, Bangalore 560 064, Professor, Dept of CSE, Dayanandsagar University, Bangalore 560 068

Retrieval Number: I32810789S319/2019©BEIESP
DOI: 10.35940/ijitee.I3281.0789S319
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

indicator for IRL as well as reuse proneness indicator.

J. Kittler [27] stated that the use of classifier ensembles with decision-level fusion can be significant for multi-class classification. Polikar et al. [28] designed an ensemble of classifiers by exploiting data from different sources and combined the base learners by means of a weighted voting scheme. AdaBoost [29] involves sub-sampling the training examples which has been applied in numerous classification problems[30].However their efficacy for highly correlated features or data seems confined. To achieve higher reliability of prediction, ensemble approach was recommended in [31, 32].

Benediktsson et.al., [33] too affirmed suitability of ensemble classifier for multi-class classification, especially derived using machine learning concept. To further augment performance prediction fusion was also recommended in [34, 35].It found that the accuracy of ensemble depends on the learner fusion for which easy averaging (i.e. maximum vote) can be a potential approach [36]. It elaborated that empirically verified ensembles aregeneralize well [37]. Briem et al. [38] used different classifiers on multiple source data having imagery of multiple spectral, Synthetic Aperture Radar (SAR) data and graphic topological information [39, 40].

Different approaches for ensemble formation were recommended in [41]. It is recommended to have predominant methods like varioushybrid strategies, classifier models, feature subsets and training set based ensemble. However lacks efficacy with multi-class classification with highly correlated features. To further augment it, integrated boosting and bagging is used to create fuzzy ensemble classifiers. Different feature sets with Principle Component Analysis (PCA) based feature selection was used in [42] to perform Rotation Forest ensemble classification. Rotation forest was used in [43] to perform to class classifications for the cancer datasets. Later, AdaBoost and Rotation Forest was combined together to develop a new ensemble RotBoost [44], where it can perform better than the classical base learners. Rotation Forest and IDE [45] was combined in [46]. SVM ensemble was recommended in [47] for spectrogram classification. As enhanced solution selection based SVM ensemble driven withclonal immunity algorithm is proposed in [48]. An ensemble NN was proposed in [49]. Partalas et al. [50] developed the preferred ensemble concept based on Focused Ensemble Selection (FES) with diversity classifiers. Amalgamating different base learners to derive an ensemble classifier has exhibited better performance than the base learner alone [51, 52]. Moderated Asymmetric Naïve Bayes classifiers (MANB) [53]homogeneous ensemble settinghave a base classifier. An ensemble pair approach is presented in [54] whichachieve better performance with respect to bagging, ensemble approaches, and ECOC. In [55], Dempster's rule of combination is used to hybridk-NN, Rocchio classifiers andSVM, and a better performance is observedwhen compared with base classifiers.

3. PROBLEM FORMULATION

Considering the significance of a robust software reusability estimation system, this paper focuses on developing an “Ensemble Classifier” algorithm with OO-software metrics and classes without imposing aging or smells. “Ensemble Classifiers” have been assessed for small readily available benchmark datasets such as Iris, Spambase, and Satellite etc; however efficacy of such algorithms with large scale and real time applications environment often remains suspicious. Thus it used 100 software modules developed using Java programming language with OO-programming concept using online software repository named “Sourceforge”.

In proposed model DLF approach with majority vote scheme has been implemented and best base classifiers propagated to the Ensemble Classifier foraugmentation of the reliability and accuracy of the reusability prediction system. The snippet of the proposed ensemble classification model based WoS software reusability prediction in Fig. 1.

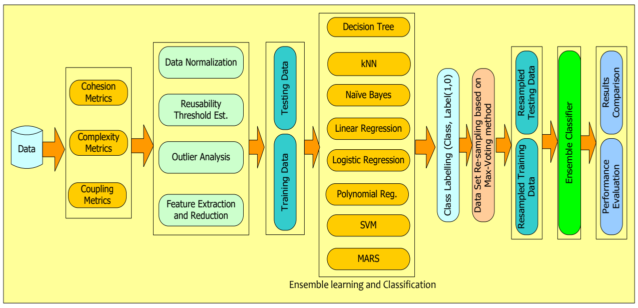

Fig. 1. Implementation Schematic of the proposed Ensemble Learning based Reusability Prediction

4. Ensemble Classifier Based Web of Service Software Reusability Prediction

The base learners and allied ensemble formation approachdiscusses the following:

4.1. *WoS software reusability data preparation*

100 OOP based software have consider WoS software, to convert into Java file using WSDL tool (Fig. 2). WSDL signifies the XML-based interface definition language characterizes web services i.e., $P = \{M_0(I_0, O_0), M_1(I_1, O_1), \dots, M_n(I_n, O_n)\}$, which can performs various functions M_i by transferring input $I_i$into output O_i. Table I shows the plotting of the various components of Java class.

Retrieval Number: I32810789S319/2019©BEIESP
DOI: 10.35940/ijitee.I3281.0789S319

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

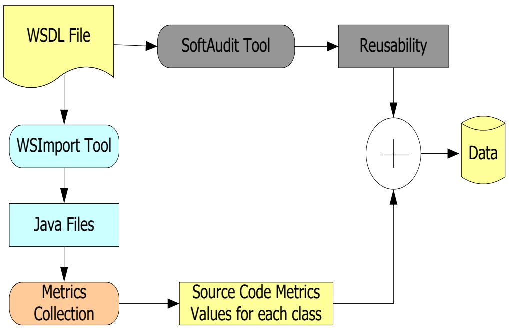

Fig. 2. Data Preparation for the proposed use case-study

Table 1. Mapping from WSDL to JAVA file.

WSDL document	JAVA document
wsdl:portType	Service Endpoint Interface (SEI) class
wsdl:portType/wsdl:operation	Approaches of the generated SEI class
wsdl:service	Service class
wsdl:service/wsdlport	getABCPort() methods of the Service class

4.2. *Data normalization and outlier analysis*

Here, each data element x_i of the class X is mapped to the allied normalized value x_i'in the range of [0, 1]. Thus, the normalized value $x_i'$is obtained as (4.1).

$$Normalized(x_i) = x_i' = \frac{x_i - min\ (X)}{max(X) - min\ (X)} \qquad (4.1)$$

4.3. *Feature selection*

A snippet of the univariate logistics regression (ULR) algorithm is depicted as follows:

4.3.1. *Uni-variate logistic regression (ulr) assisted metrics characterization*

ULR states the level of significance of each metrics towards reusability estimation of each class.

$$logit[\pi(x)] = \alpha_0 + \alpha_1 X \qquad (4.2)$$

logit[π(x)] and X state the dependent and independent variables, correspondingly. The variable π states the probability factor of significance.

$$\pi(x) = \frac{e^{\alpha_0 + \alpha_1 X}}{1 + e^{\alpha_0 + \alpha_1 X}} \qquad (4.3)$$

4.3.2. *Rough Set Analysis (RSA) based feature extraction*

RSA can be obtained in the form of classification results or reduced data elements that enable swift and precise computation. The detailed of the RSA is given as follows:

- *Step-1*: *Feature Set Selection*
  In this step, extracted CK suite metrics for each class of the software are collected.
- *Step-2*: *Feature set (Data) Discretization*
  Here, the extracted data is discretized using K-means clustering algorithm.
- *Step-3*: *Lower/Upper Approximation for all feasible data sets*
  Itestimates the lower and the upper values as the union of all comprising feature sets (Step-2).

$$\underline{B}X = \{x_i \in U | [x_i]_{Ind(B)} \subset X\} \qquad (4.4)$$

Upper approximation value signifies the union of all the encompassing sets with unit non-zero intersection with X.

$$\overline{B}X = \{x_i \in U | [x_i]_{Ind(B)} \cap X \neq 0\} \qquad (4.5)$$

- *Step-4* :*Accuracy estimation for the considered feature sets*
  Here, we have retrieved a factor stating accuracy of X in $B \subseteq A$ using (4.6).

$$\mu_B = \frac{Card(\underline{B}X)}{Card(\overline{B}X)} \qquad (4.6)$$

Cardinality of a set signifies the lower or upper approximation ofX.

- *Step-5* : *Approximated (Feature Set) Selection*

All possible sets are selected with equality of accuracy of the universal set. Later, the obtained data set with optimal cardinal possibility are chosen as the minimal set to be used for "Ensemble Learning and Classification".

4.4. Ensemble learning for software reusability prediction

Base learners have been trained over the data subsets whose outputs have been combined together to achieve better accuracy.

4.4.1. *Decision tree algorithm*

C5.0 classifier [51] used as a base classifier that exhibits recursive partitioning over the extracted OOP-metrics.

4.4.2. *Enhanced kNN Classifier*

KNN classifier applies Euclidean distance to estimate inter-attribute distance using following expression:

$$D(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_n - q_n)^2} \qquad (4.7)$$

Where$p$ and q are subjected to be compared with n features.

4.4.3. *Logistic regression*

LR algorithm constitutes a prediction scheme to examines reuse proneness of a class.

$$logit[\pi(x)] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \ldots + \beta_m X_m \qquad (4.8)$$

logit[π(x)]and x_i signifies the dependent and independent variables, correspondingly. Practically, π(x)vary from 0 to 1 to $-\infty$ to$+\infty$. LR outcome π(x) is stated as

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \ldots + \beta_m X_m}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \ldots + \beta_m X_m}} \qquad (4.9)$$

Retrieval Number: I32810789S319/2019©BEIESP
DOI: 10.35940/ijitee.I3281.0789S319
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

4.4.4. *Polynomial regression*

Univariate Polynomial Regression (UPR) for the n^{th} polynomial order:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots \ldots . + \beta_n X^n \qquad (4.10)$$

where variables X and Y states for the independent and the dependent variables and β_0, $\beta_1 \ldots \beta_n$ signify the constant and coefficients, respectively.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 \qquad (4.11)$$

For multivariate 2^{nd} order PR analysis, it is estimated on the basis of the two variables,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2 \qquad (4.12)$$

4.4.5. *Naive bayes classification*

NB allocates specified object x to class $e^* = \mathrm{argmax}_d(P(d|x))$ using Bayes′ rule.

$$P(d|x) = \frac{P(x|d)P(d)}{P(x)} \qquad (4.13)$$

P(d) signifies the probability of the parameter c before observing the data. The factor P(d|x) refers the likelihood of datax.

$$P(x|d) = \prod_{l=1}^{m} P(x_l|d) \qquad (4.14)$$

4.4.6. *Support Vector Machine (SVM)*

To perform prediction, SVM model applies the function derived in (4.15).

$$Y' = w * \phi(x) + b \qquad (4.15)$$

$\phi(x)$ states the non linear transform and Y' is retrieved by reducing the risk of regression.

$$R_{reg}(Y') = C * \sum_{i=0}^{l} \gamma(Y_i' - Y_i) + \frac{1}{2} * \|w\|^2 \qquad (4.16)$$

The parameter γ states the cost function, while C refers the penalties for error estimation.

$$w = \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) \phi(x_j) \qquad (4.17)$$

The parameters α and $\alpha^*$are Lagrange multipliers, whereα, $\alpha^* \geq 0$.

$$Y' = \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) \phi(x_j) * \phi(x) + b$$
$$= \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) * K(x_j, x) + b \qquad (4.18)$$

Where,$K(x_j, x)$ states the kernel function.

4.4.7. *Multivariate Adaptive Regression Splines (MARS)*

MARS functions of “Divide-and-conquer” policy retrieved OO-CK metrics data, which is divided into the different regions with respective regression equation.

$$Y = \sum_{i=1}^{m} C_i BF_i(X) \qquad (4.19)$$

Y states dependent and X states independent variable. The parameter C_i states the fixed coefficients and $BF_i(X)$ states the basis functions.

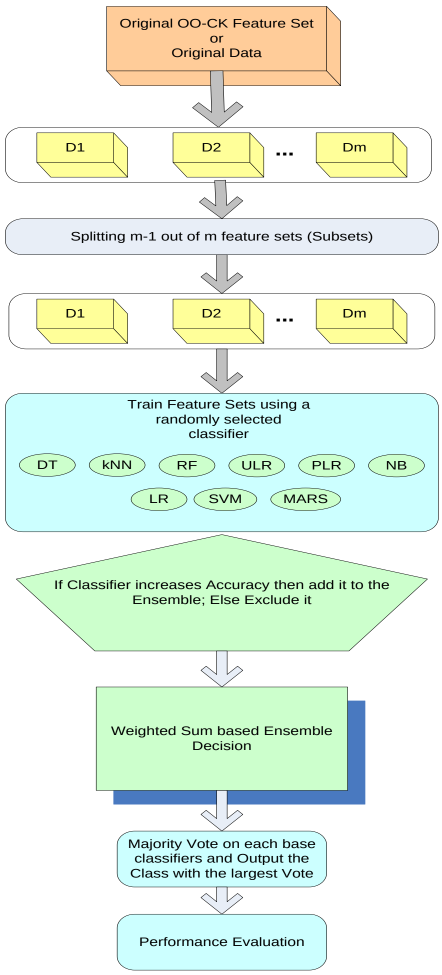

Fig. 3. Proposed Ensemble Based Classifier

4.4.8. *Ensemble structure*

The maximum weighted computational class from all base classifiers is chosen as class of prediction (4.20):

$$C(y) = argmax \sum_{j=1}^{m \times K} \alpha_j . (f_j(y) = C) \qquad (4.20)$$

α_j signifies the weight of base classifiers, $f_j(y)$ refers the reusability prediction output.

5. RESULTS AND DISCUSSION

The analysis of performance for the developed ensemble classification is as follows.

Retrieval Number: I32810789S319/2019©BEIESP
DOI: 10.35940/ijitee.I3281.0789S319

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

5.1. Performance assessment

Relative performance is examined using estimating performance of each base learners and ensemble learners. Ensemble classification concept with DLF enables higher accuracy and minimum error.

5.1.1. *Classification Accuracy Assessment*

The assessment of the developed ensemble classification model (Table 2).

Table 2. Performance Parameter

Parameter	Mathematical Expression	Definition
Accuracy	$((TN+TP))/((TN+FN+FP+TP))$	Predicted fault prone module
Precision	$\frac{TP}{(TP+FP)}$	Degree of repeated measurements under unchanged conditions having same results.
F-measure	$2.\frac{Recall.Precision}{Recall+Precision}$	The harmonic mean of the recall and precision.
Recall	$\frac{TP}{(TP+FN)}$	Identification of relevant items.
Specificity	$\frac{TN}{(TN+FP)}$	classifier identifier with negative labels.

The performance parameters and their respective values are given in Table 3.

Table 3. Performance values

Techniques	Accuracy(%)	Precision(%)	F-Measure(%)	Recall(%)	Specificity(%)
Decision Tree	91.31	90.01	91.65	91.02	82.64
Naïve Bayes	81.40	95.45	88.59	81.78	71.00
kNN	88.43	87.31	89.93	88.99	87.74
Linear Regression	85.71	84.60	86.82	81.30	84.24
Polynomial Regression	89.01	90.48	92.31	89.95	91.04
Logistic Regression	87.91	90.32	91.73	90.16	88.04
SVM (Polynomial)	93.41	94.97	95.52	91.04	91.27
MARS	88.12	93.20	94.20	89.70	96.00
Proposed Ensemble	**97.02**	**95.64**	**96.18**	**94.93**	**97.01**

5.1.2. *Error Profiling*

Ensemble learning intends to reduce error of prediction by amalgamating different classifiers and their respective performances over the original feature data.

5.1.2.1 Mean absolute error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|Y_i' - Y_i|) \qquad (5.1)$$

where Y_i'refers the calculated output, while Y_i states for the expected value.

5.1.2.2 Mean magnitude of the relative error (MMRE)

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i' - Y_i|}{Y_i} \qquad (5.2)$$

To deal with overflow conditions ($Y_i = 0$), 0.01is added to denominator. Thus, MMRE i

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i' - Y_i|}{Y_i + 0.01} \qquad (5.3)$$

5.1.2.3 Standard Error of the Mean (SEM)

Mathematically, SEM is presented as (5.4).

$$SEM = \frac{\sigma}{\sqrt{N}} \qquad (5.4)$$

where σ and $N$are standard deviation and total number of samples, respectively.

Table 4 Error performance by the different base learners and the ensemble classifier

Techniques	MAE	MMRE	RMSE	SEM
Decision Tree	0.0990	0.7000	0.0200	0.0900
Naïve Bayes	0.1557	0.8623	0.1209	0.0276
kNN	0.2102	0.7432	0.1953	0.1003
Linear Regression	0.2390	0.8418	0.2052	0.1167
Polynomial Regression	0.1859	0.8016	0.0591	0.962
Logistic Regression	0.1968	0.8073	0.542	0.1331
SVM (Polynomial)	0.1904	0.8001	0.0245	0.1090
MARS	0.2040	0.8210	0.2002	
Proposed Ensemble	0.0027	0.0631	0.0091	0.0087

CONCLUSIONS

This paper proposes anensemble learning concept using machine learning. WoS data was obtained from 100 software developments using Object Oriented Programming (OOP) concept. The software data is processed with WSDL followed by Chidamber and Kamrer (CK) metrics estimation. The extracted CK-metrics data is processed for Min-Max normalization followed by outlier detection that avoided premature convergence issues significantly. Furthermore, RSA feature extraction technique enabled selection of only relevant and significant attributes related to software (class) reusability. Considering other base classifiers such as neural network and its variant, shows these approaches suffer from local minima and convergence, and therefore evolutionary computing based learning can be explored to alleviate

Retrieval Number: I32810789S319/2019©BEIESP
DOI: 10.35940/ijitee.I3281.0789S319

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

such issues. This as a result could yield more efficient and reliable ensemble classification.

REFERENCES

1. G. Caldiera, and V.R. Basili, “Identifying and qualifying reusable software components,” IEEE Software, vol. 24, pp. 61-70, 1991.
2. I. Sommerville, “Software Engineering”. 9th Edition, Addison-Wesley, New York, 2011.
3. G. Singh, “Metrics for measuring the quality of object-oriented software,” ACM SIGSOFT Software Engineering Notes, vol. 38, pp. 1-5, 2013.
4. M. M. Stapic, “Reusability metrics of software components: Survey”, conference paper · September, 2015.
5. S. Srivastava and R. Kumar, "Indirect method to measure software quality using CK-OO suite," Intelligent Systems and Signal Processing (ISSP), International Conference on, Gujarat, pp. 47-51, 2013.
6. B. M. Goel, Pradeep Kumar Bhatia “ Analysis of reusability of object-oriented system using CK metrics”, International Journal of Computer Applications, vol. 60, no.10, 2012.
7. N.S. Bakar “The analysis of object-oriented metrics in C++ programs”, Lecture Notes on Software Engineering, vol. 4, no. 1, 2016
8. A. T. Mohammad “Metric suite to evaluate reusability of software product line”, International Journal of Electrical and Computer Engineering (IJECE), vol. 4, no. 2, pp. 285-294, 2014.
9. L. Nanni, S. Brahnam, S. Ghidoni, and A. Lumini, “Toward a General-PurposeHeterogeneous Ensemble for Pattern Classification,” Computational Intelligence and Neuroscience, vol. 2015.
10. M. Barghash, “An effective and novel neural network ensemble for shift pattern detection in control charts,” Computational Intelligence and Neuroscience, vol. 2015, Article ID939248, 2015.
11. G. Tsoumakas, L. Angelis, I. Vlahavas, Selective fusion of heterogeneous classifiers, Intelligent Data Analysis Vol. 9 (6), pp. 511–525, 2005
12. R. R. Brooks, P. Ramanathan; A. M. Sayeed, “Distributed Target Classification and Tracking in Sensor Networks, Proceedings of the IEEE Vol. 91, pp. 1163-1171, 2003.
13. C. Dong-Sheng, et al., The boosting: a new idea of building models, Elsevier, Amsterdam, PAYS-BAS, 2010, vol. 100, pp.1–11.
14. K. M. Ting, Z.Zheng, A study of Adaboost with Naïve Bayesian classifiers: weakness and improvement, Computational Intelligence Vol. 19(2), pp. 186–200, 2003.
15. H. Singhani, R. P. Suri “Testability assessment model for object oriented software based on internal and external quality factors”, Global Journal of Computer Science and Technology: C Software & Data Engineering, vol. 15, no.5, 2015.
16. M. Mijac, Z. Stapic, “Reusability metrics of software components: Survey”, Central European conference on Information and Intelligent system conference paper · September, 2015.
17. S. Srivastava and R. Kumar, "Indirect method to measure software quality using CK-OO suite," Intelligent Systems and Signal Processing (ISSP), International Conference on, Gujarat, pp. 47-51, 2013.
18. B.M. Goel, and P.K. Bhatia, “Analysis of reusability of object-oriented system using CK metrics,” International Journal of Computer Applications, Vol.60, No.10, pp.0975–8887, 2012.
19. L.H. Rosenberg, and L.E. Hyatt, “Software quality metrics for object-oriented environments,” Crosstalk Journal, vol. 10, pp. 1-16, 1997
S.R. Chidamber, and. C. F. Kemerer, “A metrics suite for object oriented design,” IEEE Transactions on Software Engineering, vol. 20, pp. 476-493, 1994.IEEE PressPiscataway, NJ, USA.
20. R. Shatnawi, W. Li, J. Swain, and T. Newman, “Finding software metrics threshold values using roc curves,” Journal of Software Maintenance and Evolution: Research and Practice, vol. 22, pp. 1-16, 2010.John Wiley & Sons, Inc.New York, NY, USA.
21. P. Neelamdhab, S. Satapathy, R. Singh, (2017). Utility of an Object Oriented Reusability Metrics and Estimation Complexity. *Indian Journal Of Science And Technology*, *10*(3).
22. Normi Sham Awang Abu Bakar “The analysis of object-oriented metrics in C++ programs”, Lecture Notes on Software Engineering, Springer ,vol. 4, no. 1, 2016.
23. S. I. Zahara, M. Ilyas, and T. Zia, "A study of comparative analysis of regression algorithms for reusability evaluation of object oriented based software components," Open Source Systems and Technologies (ICOSST), International Conference on, Lahore, pp. 75-80, 2013.
24. Torkamani M. A. “Metric suite to evaluate reusability of software product line”, International Journal of Electrical and Computer Engineering (IJECE), vol. 4, no. 2, pp. 285-294, 2014.
25. A. Aloysius, and K. Maheswar, “A review on component based software metrics”, Intern. J. Fuzzy Mathematical Archive, vol. 7, no. 2, pp. 185-194, 2015.ISSN: 2320 –3242 (P), 2320 –3250 (online)
26. E.S. Cho, M.S. Kim, and S.D. Kim, (2001) “Component metrics to measure component quality,” Proceedings of the 8th Asia Pacific Software Engineering Conference (APSEC), Macau, vol. 4-7, pp. 419-426, 2001.
27. J. Kittler, Multi-Sensor Integration and Decision Level Fusion. *Proc. DERA/IEE Workshop Intelligent Sensor Processing* 2001, 1-6.
28. R. Polikar, D. Parikh; Shreekanth; Mandayam. Multiple Classifier Systems for Multisensor Data Fusion. *SAS 2006 - IEEE Sensors Applications Symposium* 2006, 180-184.
29. Schapire, R. E. A Brief Introduction to Boosting. Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence 1999, 1401-1406.
30. T. G. Dietterich, Machine learning research: Four current directions. *AI Magazine* 1997, *18*, 97- 136.
31. M. Barghash, “An effective and novel neural network ensemble for shift pattern detection in control charts,” *Computational Intelligence and Neuroscience*, vol. 2015, Article ID939248, 2015.
32. L. Nanni, S. Brahnam, S. Ghidoni, and A. Lumini, “Toward a General-PurposeHeterogeneous Ensemble for Pattern Classification,” Computational Intelligence and Neuroscience, vol. 2015.
33. J. A Benediktsson, J. Chanussot, and M. Fauvel, Multiple classifier systems in remote sensing: from basics to recent developments. In: M. Haindl, J. Heidelberg, Germany: Springer, 501–512, 2007.
34. F. Roli, G. Giacinto, G. Vernazza, Methods for designing multiple classifier systems. Proceedings of the second international workshop on multiple classifier systems.

Retrieval Number: I32810789S319/2019©BEIESP
DOI: 10.35940/ijitee.I3281.0789S319

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Cambridge, UK, 78–87, 2001.

35. Banfield, R., et al., A comparison of decision tree ensemble creation techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29, 173–180, 2007.
36. L. Hansen, P. Salamon, Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, 993–1001, 1990.
37. D. Opitz, J. Shavlik, J., Actively searching for an effective neural-network ensemble. Connection Science, 8 (3/4), 337–353, 1996.
38. G. J. Briem, G.J., Benediktsson, J.A., J. R. Sveinsson, Multiple classifiers applied to multisource remote sensing data. IEEE Transactions on Geoscience and Remote Sensing, 40 (10), 2291–2299, 2002.
39. B. Waske, J.A. Benediktsson, Fusion of support vector machines for classification of multisensor data. IEEE Transactions in Geoscience and Remote Sensing, 45, 3858–3866, 2007.
40. L.I. Kuncheva, Combining classifiers: soft computing solution, in: S.K.A. Pal (Ed.), Pattern Recognition: From Classical to Modern Approaches, World Scientific, Singapore, 2001, pp. 427–451.
41. J. Canul-Reich, L. Shoemaker, L.O. Hall, Ensembles of fuzzy classifiers, in: IEEE International Fuzzy Systems Conference, pp. 1–6, 2007.
42. J.J. Rodriguez, L.I. Kuncheva, Rotation forest: a new classifier ensemble method, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1619–1630.
43. K. H. Liu, D. S. Huang, Cancer classification using rotation forest, Computers in Biology and Medicine 38, 601–610, 2008.
44. Chun-Xia Zhang, Jiang-She Zhang, RotBoost: a technique for combining rotation forest and AdaBoost, Pattern Recognition Letters 29, pp. 1524–1536, 2008.
45. K. Tumer, N.C. Oza, Input decimated ensembles, Pattern Analysis Application, vol. 6, pp. 65–77, 2003.
46. L. Nanni, A. Lumini, Ensemble generation and feature selection for the identification of students with learning disabilities, Expert Systems with Applications 36, pp.3896–3900, 2009.
47. Loris Nanni, Alessandra Lumini, Input decimated ensemble based on neigh- borhood preserving embedding for spectrogram classification, Expert Systems with Applications 36,pp. 11257–11261, 2009.
48. X. Zhang, S. Wang, T. Shan, L.C. Jiao, Selective SVMs ensemble driven by immune clonal algorithm, in: Rothlauf, F. (Ed.) Proceedings of the EvoWork- shops, Springer, Berlin, pp. 325–333, 2005.
49. Z.H. Zhou, J. Wu,W. Tang, Ensembling neural networks: many could be better than all, Artificial Intelligence 137 (1–2), 239–263, 2002.
50. I. Partalas, G. Tsoumakas, I. Vlahavas, Focused ensemble selection: a diversity-based method for greedy ensemble selection, in: Proceedings of the 18th International Conference on Artificial Intelligence, pp. 117–121, 2008.
51. S. Bothorel, BB. Meunier, S. A. Muller, Fuzzy logic based approach for semilogical analysis of microcalcification in mammographic images. Int J Intell System, 12:819–848, 1997.
52. AT. Stavros, D. Thickman, CL.Rapp, MA. Dennis, SH Parker, GA Sisney. Solid breast nodules: use of sonography q1q`to distinguish between benign and malignant lesions. Radiology; 196:123–134, 1995.
53. YS. Dong, KS Han, "A comparison of several ensemble methods for text categorization," Services Computing, (SCC 2004). Proceedings. 2004 IEEE International Conference on, pp. 419-422, 2004.
54. Y. Liu, G. Jaime, Carbonell, and Rong Jin. ECML, volume 2837 of Lecture Notes in Computer Science (Springer), pp. 277-288, 2003.
55. Y. Bi, D. Bell, H. Wang, G. Guo, J. Guan, Combining Multiple Classifiers Using Dempster's Rule For Text Categorization. Appl. Artif. Intell. Vol. 21, 211-239, 2007.

Retrieval Number: I32810789S319/2019©BEIESP
DOI: 10.35940/ijitee.I3281.0789S319

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication