VOL. 17, NO. 5, MARCH 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

BEHAVIORAL-BASED KERNEL NEUTROSOPHIC CLUSTERING FOR HETEROGENEOUS CROSS-PROJECT DEFECT PREDICTION

N. Kalaivani[1] and R. Beena[2]
[1]Department of Computer Science, Kongunadu Arts and Science College, Coimbatore, Tamil Nadu, India
[2]Department of Information Technology, Sri Ramakrishna College of Arts and Science, Coimbatore, Tamil Nadu, India
E-Mail: kalaivhani@gmail.com

ABSTRACT

Software defect prediction is essential in software development and maintenance, a highly demanded quality of service. Heterogeneous defect prediction is the most appropriate method for real-time datasets. The heterogeneous metrics of cross-projects are used to predict in many existing models, but the presence of outlier and noisy datasets is not considered an essential factor; thus, the standard prediction models face challenges in producing more accurate results. This paper focuses on handling the impreciseness and vagueness in treating noisy and outliers in software defect prediction datasets. This is accomplished by adapting bipartite ranking-based feature ranking, which converts the target attribute size same as the source attribute size and the feature selection by selecting the top attributes. The noisy and outlier is handled by kernel neutrosophic clustering by introducing the degree of truthiness, indeterminacy and falsity. Finally, Grey Wolf Optimization enhances the heterogeneous cross-project prediction process by selecting the significant centroids in kernel neutrosophic clustering unlabeled instances. This work used six different heterogeneous datasets for software defect prediction, and the results explore that the proposed model performs better and prominently increases the prediction rate.

Keywords: heterogeneous cross-project, bipartite ranking, kernel neutrosophic clustering, grey wolf optimization, defect prediction.

INTRODUCTION

As there is a tremendous increase in demand for using software products and their functionalities, maintaining the quality of the software becomes a challenging problem in the field of software engineering. Data mining models greatly help maintain software quality assurance by predicting the software defects that may occur in software entities using the historical data during their development stage. Conventional Software defect prediction models initially develop a classification method using the adequate historical labeled software entity details from a project under consideration [1]. Then it uses that model to predict the presence or absence of defects for the newly developed software entities within the same project. Predicting the labeled entities within the same software is known Within-Project Defect Prediction (WPDP). But WPDP is not feasible for most real-time projects because adequate training data will not always be possible while handling new or immature projects [2].

In the Cross-project, Defect Prediction (CPDP) develops the classifier corresponding to the labeled defect data from other software projects, known as source projects, to predict the defect labels from a target project [3]. Unfortunately, the problem while using CPDP is that this will not be feasible when the projects with different attributes are involved in the prediction process. Programming languages may differ most of the time, and the attributes are gathered from different levels using several tools. Hence, CPDP fails to predict the software defect on two different projects with different attributes. This kind is referred to as Heterogeneous Cross-project Defect Prediction (HCPDP). As the attributes of two different projects don't have communal elements, CPDP or WPDP cannot be used.

This paper focuses on Heterogeneous Cross-project Defect Prediction, which constructs a forecasting approach using source and target software projects with different metrics. The main idea of HCPDP is to develop a projective matrix among heterogeneous source and target projects which translates the source pace to the target space or vice versa. After converting them into the same space, the classifier predicts the target project defect. First, feature selection is applied to remove useless attributes in the source project. Then, similarity computation is done to match the chosen attributes in the source to the target attributes. The attributes with less matching score are eliminated based on the predetermined threshold.

This research work develops an unsupervised learning model by applying Kernel Neutrosophic C Means clustering enhanced by gray wolf optimization for heterogeneous cross-project defect prediction.

RELATED WORK

Yin *et al* [4] developed a heterogeneous defect prediction using multi-source projects for defect prediction. They created a projective matrix among source and target projects by transferring learning and making both source and target distribution similar.

Vashisht, Rohit [5] designed an empirical modeling phase involving feature extraction on heterogeneous cross-project defect prediction. They applied feature selection to eliminate the redundant feature in the dataset. Chi-Square is applied for feature selection, and principal component analysis is used for feature extraction. They used the classification model to predict software defects.

Yang *et al* [6] aim to understand a common subspace for all domains using a heterogeneous graph attention network model. This model performs semantic propagation over the related instances among multiple datasets. Aligning source and target pseudo labeled centroid in graph attention network are used for prediction.

Jahanshahi *et al* [7] devised an ensemble voting approach for heterogeneous defect prediction of multiple datasets. They performed transfer learning used to convert one domain space to another domain. This model also reveals the feasibility of using HDP in real datasets.

Nam *et al* [8] used metric matching with a large enough sample of source and target projects for prediction among heterogeneous software projects. They used different categories of data sets to construct a mathematical model-based heterogeneous defect prediction.

Ni *et al* [9], in their work, feature selection is achieved using clustering of hybrid data using density-based clustering. This feature selection greatly helps to correlate features into clustering more effectively. Next, significant features are selected from each clustering, and they designed three ranking policies. Based on the selected features, prediction is made on real-world software projects.

Kernel Neutrosophic Clustering Enhanced by Grey Wolf Optimization (KNC-GWO) for Heterogeneous Cross-Project Defect Prediction

This research work concentrates on unsupervised modeling for heterogeneous cross-project defect prediction. In this model, the target dataset is involved in feature reduction, which the bipartite ranking model does. The six different datasets are used in this work with different metrics collected from five different software groups [14, 15, 16]. The reduced feature set of the target dataset is clustered with the source dataset using neutrosophic C means clustering; this is optimized by applying metaheuristic nature-inspired algorithm gray wolf optimization. This method detects the defect in heterogeneous cross-project with unlabeled instances. The overall architecture of the proposed KNC-GWO for Heterogeneous Cross-project Defect Prediction is shown in Figure-1.

METHODOLOGY

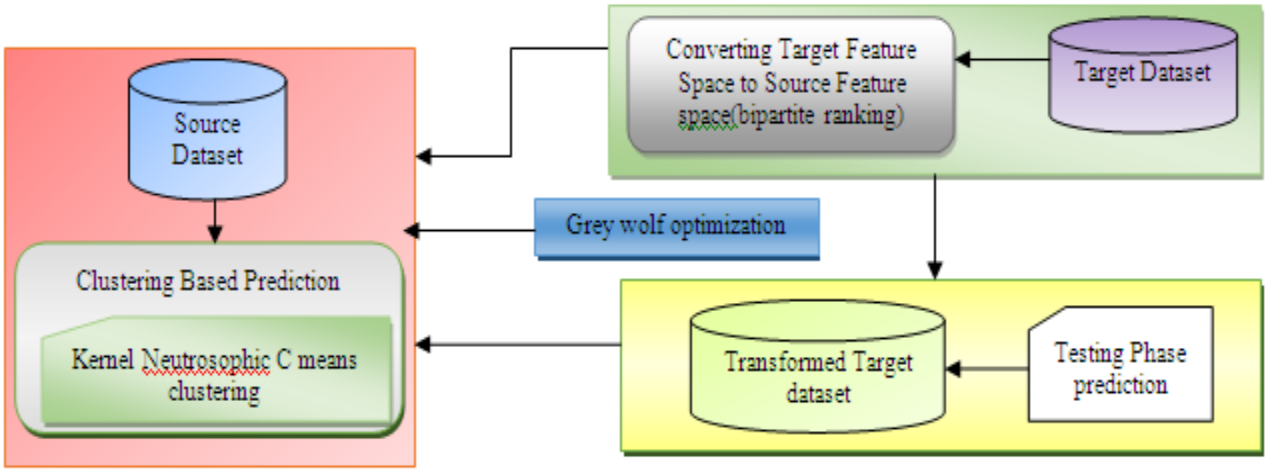

Figure-1. Overview of KNC-GWO for HCPDP.

Bipartite Ranking Based Feature Reduction in the Target Dataset

Machine learning-based ranking can be applied to reinforcement learning, semi-supervised and supervised learning. It involves developing ranking models for the prediction process. The training dataset comprised of item lists with partial orders mentioned among items in a given list; the order may be either ordinal or numeric depending on their relevancy and irrelevancy of each item. The purpose of using ranking methods is to rank the items in the list based on some critical criteria.

This research work uses a bipartite ranking model for translating the target project space to the source project space to perform effective heterogeneous cross-project defect prediction.

Bipartite ranking is a kind of ranking function which learns about the defected or non-defected instances of labeled software datasets. While bipartite ranking is applied to a set of unlabeled datasets, it performs ranking function to establish a total order is which defective instances precede non-defective ones. The goal of the bipartite ranking is to learn a ranking function br: X xX→ {−1, 0, 1}, where br(x,x') = 1 means it is considered as x is ranked greater than x'. If br(x, x') = 0 then it indicates that there is a tie between x and x' [10]. Rank loss is used to measure the accuracy of bipartite ranking function based on the probability of incorrectly ordering in terms of one positive and one negative selected using Prob(x,y). It is formulated as follows:

$$Loss_{rnk}(r) = \text{Prob}(\text{br}(\text{x},\text{x}') = -1 \,|\text{y} > y') + \frac{1}{2} prob(br(x,x') = 0|y > y')$$
$$= \int \frac{|1-\text{br}(\text{x},\text{x}')|}{2} \text{dProbr}(\text{x}|\text{y} = 1)\ \text{dProb}(\text{x}|\text{y} = 1)\text{dProb}(\text{x}'|\text{y}' = -1)$$

Algorithm for Bipartite Feature Elimination

Input: Software dataset

Output: Reduced Feature set of Software Dataset

Procedure

1. Calculate the Labels weights and compute the correlation matrix using weight assigned to the label
2. Compute the Correlation Matrix
3. Compute sum of distance among attributes of software dataset by applying weight correlation distance gain correlation distance vector

EQD (att_i,att_j) = $\sqrt{\sum_{H=1}^{m}(att_i - att_j)^2}$

Where EQD (att_i,att_j) computes the euclidean distance among attributes att_i and att_j.. m refers to number of labels

4. By applying Smooth Rank algorithm find the potential attributes with maximum matching to label.

a. For each attribute att_i

i. Construct kernel approximation kg_1^i,kg_2^i for the density of the class BR^i=dm(att_i)

ii. For each instance br∈BR^i compute

$$q_i(\mathrm{br}) = \frac{\mathrm{kg}_1^{\mathrm{i}}(\mathrm{br}) - \mathrm{kg}_1^{\mathrm{i}}(\mathrm{br})}{\pi_1.\,\mathrm{kg}_1^{\mathrm{i}}(\mathrm{br}) + \pi_2.\,\mathrm{kg}_2^{\mathrm{i}}(\mathrm{br})}$$

Where π_1 and π_2 are the class frequencies of class 0 and 1 respectively

iii. Loss approximation function $q_i(br)$ is used for constructing marginal predicators $\widetilde{q_i}(br)$

iv. Compute predictors weight wt_i using the correlation

b. Compute ranking function as formulated in the equation below

$$RF(att) = \frac{\sum_{i:att_i \neq NA} wt_i.\widetilde{q_i}(att_i)}{\sum_{i:att_i \neq NA} wt_i}$$

5. Sort the attributes in each matching based on correlation distance vector

Kernel Based Neutrosophic Clustering for Heterogeneous Cross-Project Defect Prediction

Neutrosophic logic is represented in the triplet from represented as degree of membership of truthiness, falsity and indeterminacy [11]. It is denoted as

NL=<Trt, Ind, Fls>

Where Trt, Ind, Fls may be standard or non-standard values between]$^-0$ 1^+ [.The Neutrosophic logic handles the impreciseness, uncertainty and vagueness.

A very essential task in machine learning and data mining is clustering technique. The input data is categorized based on the similarity measure applied to them. In standard clustering namely, K-means and Fuzzy C-means all the instances in the dataset are treated with equal importance without considering outlier and noisy instances. But in real datasets, like software defect dataset may have noise and outlier that required to be resolute. This issue can be overcome by neutrosophic clustering with handles both outliers and noises. It deliberates both belongingness degrees to determinate and indeterminate clusters with new objective function. The kernel objective function [12] is applied to handle the high dimensional attribute space, which is defined in the following equation

$$\mathrm{OBJ}_{\mathrm{KNCM}}(\mathrm{Trt},\mathrm{Ind},\mathrm{Fls},\mathrm{C}) = \sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{j}=1}^{\mathrm{C}}\left(\rho_1\mathrm{Trt}_{\mathrm{ij}}\right)^{\mathrm{m}}\left\|\Delta(\mathrm{x}_\mathrm{i}) - \Delta\left(\mathrm{c}_\mathrm{j}\right)\right\|^2 + \sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\rho_2\mathrm{Ind}_{\mathrm{ij}}\right)^{\mathrm{m}}\left\|\Delta(\mathrm{x}_\mathrm{i}) - \Delta(\mathrm{c}_{\mathrm{imax}})\right\|^2 + \delta^2\sum_{\mathrm{i}=1}^{\mathrm{N}}(\rho_3\mathrm{Fls}_\mathrm{i})^{\mathrm{m}}$$

and $\left\|\Delta(\mathrm{x}_\mathrm{i}) - \Delta\left(\mathrm{c}_\mathrm{j}\right)\right\|^2 = \mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{x}_\mathrm{j}\right) - 2\mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{c}_\mathrm{j}\right) + \mathrm{kn}(\mathrm{c}_{\mathrm{i}},\mathrm{c}_\mathrm{j})$

where $kn\left(x_i - x_j\right) = \Delta^{Trt}(x_i)\Delta(x_i)$ explores inner product, if kernel function is a gaussian function then $kn\left(x_i, x_j\right)$ and $kn(c_i,c_j)$ will be one. The new objective function will be

$$\mathrm{OBJ}_{\mathrm{KNCM}}(\mathrm{Trt},\mathrm{Ind},\mathrm{Fls},\mathrm{C}) = \sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{j}=1}^{\mathrm{C}}\left(\rho_1\mathrm{Trt}_{\mathrm{ij}}\right)^{\mathrm{m}}(1 - \mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{c}_\mathrm{j}\right)) + \sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\rho_2\mathrm{Ind}_{\mathrm{ij}}\right)^{\mathrm{m}}(1 - \mathrm{kn}(\mathrm{x}_\mathrm{i},\mathrm{c}_{\mathrm{imax}})) + \delta^2\sum_{\mathrm{i}=1}^{\mathrm{N}}(\rho_3\mathrm{Fls}_\mathrm{i})^{\mathrm{m}}$$

$$\mathrm{c}_\mathrm{j} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\rho_1\mathrm{Trt}_{\mathrm{ij}}\right)^{\mathrm{m}}\mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{x}_\mathrm{j}\right))}{\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\rho_1\mathrm{Trt}_{\mathrm{ij}}\right)^{\mathrm{m}}}$$

The truthiness membership function is formulated as:

$$\mathrm{Trt}_{\mathrm{ij}} = \frac{\rho_2,\rho_3\mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{c}_\mathrm{j}\right)^{-\left(\frac{2}{\mathrm{m}-1}\right)}}{\sum_{\mathrm{j}=1}^{\mathrm{c}}\mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{c}_\mathrm{j}\right)^{-\left(\frac{2}{\mathrm{m}-1}\right)} + \mathrm{kn}(\mathrm{x}_\mathrm{i},\mathrm{c}_{\mathrm{imax}})^{-\left(\frac{2}{\mathrm{m}-1}\right)} + \delta^{-\left(\frac{2}{\mathrm{m}-1}\right)}}$$

The indeterministic function is

$$\mathrm{Ind}_{\mathrm{ij}} = \frac{\rho_1,\rho_3\mathrm{kn}(\mathrm{x}_\mathrm{i},\mathrm{c}_{\mathrm{imax}})^{-\left(\frac{2}{\mathrm{m}-1}\right)}}{\sum_{\mathrm{j}=1}^{\mathrm{c}}\mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{c}_\mathrm{j}\right)^{-\left(\frac{2}{\mathrm{m}-1}\right)} + \mathrm{kn}(\mathrm{x}_\mathrm{i},\mathrm{c}_{\mathrm{imax}})^{-\left(\frac{2}{\mathrm{m}-1}\right)} + \delta^{-\left(\frac{2}{\mathrm{m}-1}\right)}}$$

The falsity membership function is

$$\mathrm{FLS}_{\mathrm{ij}} = \frac{\rho_1,\rho_2(\delta)^{-\left(\frac{2}{\mathrm{m}-1}\right)}}{\sum_{\mathrm{j}=1}^{\mathrm{c}}\mathrm{kn}\left(\mathrm{x}_\mathrm{i},\mathrm{c}_\mathrm{j}\right)^{-\left(\frac{2}{\mathrm{m}-1}\right)} + \mathrm{kn}(\mathrm{x}_\mathrm{i},\mathrm{c}_{\mathrm{imax}})^{-\left(\frac{2}{\mathrm{m}-1}\right)} + \delta^{-\left(\frac{2}{\mathrm{m}-1}\right)}}$$

Grey Wolf Optimization for Centroid Selection

In this paper to improve the kernel neutrosophic clustering, initial centroid selection is done by grey wolf optimization algorithm instead of performing it in a random manner. As the grey wolf is a Canidae family, they are at the top position of food chain [13]. They live in a pack, which comprised a male and female leader known as alphas. Decision making regarding sleeping place, hunting and wake up time are done by alpha wolves. The next level of this pack is beta wolf with may be male or female selected among the alpha wolves which is very old. The beta wolf must respect the alpha wolf and pass the commands to the other lower level wolves and it plays an important role in this pack as an advisor to alpha and acts as a disciplined for the entire group. The lowest ranking wolves in this pack are called as omega which acts as scapegoat; they must obey all the dominant wolves.

The behavior of the gray wolves used to hunt their prey is known as encircling behavior this is adapted in kernel neutrosophic clustering to select the centroids as prey.

Procedure: Grey wolf Optimization algorithm for selecting the centroids

Initialize the population of grey wolf GWi
Set the parameters a, B and E
Compute the fitness of each search agent
GWα← the Best search agent
GWβ← Second Best search agent
GWδ← third Best search agent
While n <max_iter
For each search agent
Update the position of the current search agent

$D\alpha = |E_1.GW_\alpha - Gw|$, $D_\beta = |E_2.GW_\beta - Gw|$, $D_\delta = |E_3.GW_\delta - Gw|$

$GW_1 = GW_\alpha - B_1.D\alpha$, $GW_2 = GW_\beta - B_2.D_\beta$, $GW_3 = GW_\delta - B_3.D_\delta$

$$GW(n+1) = \frac{GW_1 + GW_2 + GW_3}{3}$$

End for
Update a, B and E
Compute the fitness of all search agents
Update the GWα, GW$\beta$and GWδ
N = n+ 1
End while
Return GWα

Algorithm for Grey Wolf Optimization based Kernel Neutrosophic C means Clustering based Heterogeneous cross-project Defect prediction

Input: Heterogeneous software dataset

Output: Categorize defect and non-defect software

Begin
Steps

1. Initialize $Trth^{(0)}$, $Ind^{(0)}$ and $Fls^{(0)}$
2. Initialize C, m,δ, ρ_1, ρ_2, ρ_3
3. Initialize the centroids using GWO
   Call algorithm Grey wolf optimization
4. Select Kernel function and its associated parameters
5. Compute the centers $C^{(k)}$ at k step

$$c_j = \frac{\sum_{i=1}^{N}(\rho_1 Trt_{ij})^m kn(x_i, x_j))}{\sum_{i=1}^{N}(\rho_1 Trt_{ij})^m}$$

6. Calculate the C_{imax} using the clusters centers with the first largest and second largest value of Trth

$$c_{imax} = \frac{C_{fi} - C_{si}}{2}$$

Where $f_i = \text{argmax}_{j=1,2,\dots C}(Trt_{ij})$, $s_i = \text{argmax}_{j \neq f_i \cap j=1,2,\dots C}(Trt_{ij})$

7. Update $Trth^{(k)} to\ Trth^{(k+1)}, Ind^{(k)} to\ Ind^{(k+1)}\ and\ Fls^{(k)} to\ Fls^{(k+1)}$
8. If $\left|Trth^{(k)} - Trth^{(k+1)}\right| < \varepsilon$ then terminate the algorithm else go to step 5
9. Assign each instance into the class with the largest Trthm = [trth,Ind,Fls] x(i) $\in$ kth class

End

RESULTS AND DISCUSSIONS

This section discusses about the detailed performance analysis of proposed kernel Neutrosophic Clustering enhanced using Grey Wolf Optimization (KNC_GWO) for Heterogeneous Cross-project Defect Prediction (HCPDP). The KNC-GWO is implemented using MALTAB Software. This work used five different software groups and six defect datasets are used for HCPDP [14, 15, 16]. The detailed description is shown in the Table-1.

Table-1. Six different datasets of five different software groups.

Group	Dataset	No of Instances	No of Metrics	Prediction Attribute
AEEEM	EQ	324	61	class
MORPH	Velocity-1.4	196	20	class
SOFTLAB	Ar1	121	29	function
NASA	CM1	344	37	function
	PC2	1585	36	function
ReLink	Apache	194	26	file

Table-1 explores AEEM software group with EQ dataset which has 61 metrics as attributes and 324

instances. From MORPH software group, Velocity-1.4 dataset is selected which comprised of 20 metrics. The NASA software group has two different datasets CM1 and PC2 with 37 and 36 attributes respectively, The SOFTLAB software group holds Ar1 dataset with 29 attributes and finally ReLink software group contains Apache dataset with 26 metrics as attributes.

The Figure-2 explores the performance metric of correctly and incorrectly clustered instances of two different dataset Ar1 and CM1. The bipartite ranking is used to perform feature reduction on the target dataset CM1 to work with Ar1. The presence of buggy and clean instances is categorized using three clustering models KMC, FCM and proposed KNC-GWO. The highest percentage of correctly clustered is achieved by the proposed kernel neutrosophic clustering enabled using grey wolf optimization greatly handles the instances of outliers and noisy instances by defining each instance with the degree of belongings and non-belongings. Where the standard KMC and FCM doesn't considered the noisy and outlier instances during clustering and they result in highest error rate of software defect prediction whereas KNC-GWO produce less error rate while clustering the buggy and clean datasets.

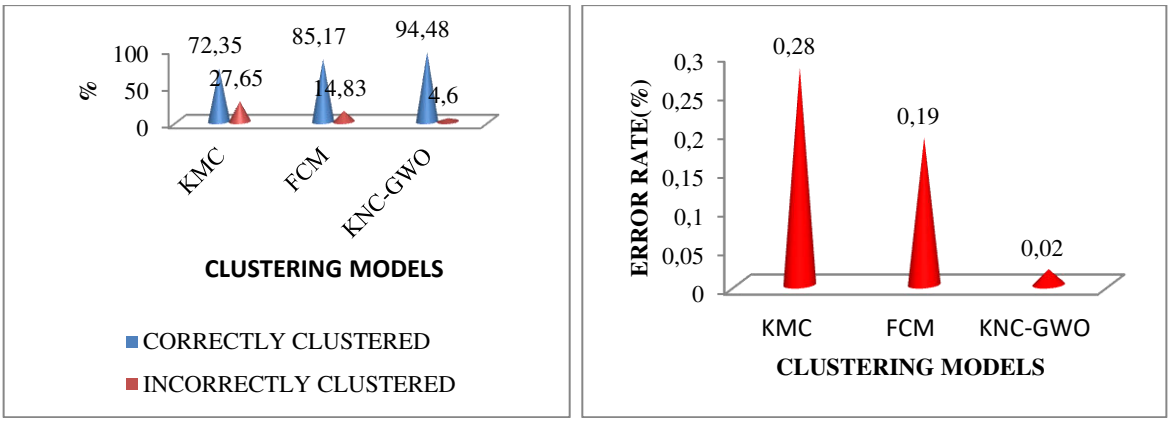

Figure-2. Performance comparison of dataset Ar1-SOFTLAB and dataset CM1-NASA.

Figure-3 depicts the performance of three different clustering models involved in prediction of software defects in HCPD based on correctly, incorrectly clustered rate and Error rate obtained by them. Here, PC2 dataset belong to NASA software group and Apache dataset belongs to reLink software group. These two heterogeneous projects are converted into single dataset by first converting the attribute size of the target dataset using bipartite feature elimination. These two datasets similarity is measured and clustered using kernel neutrosophic clustering. Thus, it handles the noisy and outlier instances present in the heterogeneous cross-project dataset. The proposed KNC-GWO achieves optimal clustering rate compared to the k means and fuzzy c means. The error rate of the proposed KNC-GWO is very less compared to other two clustering models, because they follow the random cluster centroid selection in the initial stage, whereas KNC-GWO uses the knowledge of grey wolf optimization to select the most promising instances as cluster centroids and thus the software defect prediction process achieves best result.

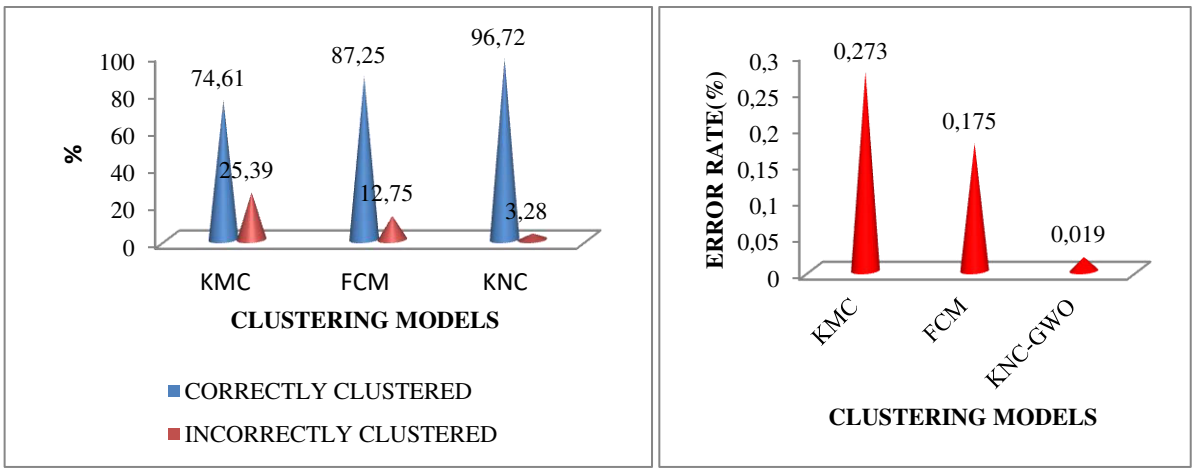

Figure-3. Performance comparison NASA group dataset PC2 and RELINK group dataset Apache.

Software defect prediction in heterogeneous cross-project datasets namely velocity 1.4 from MORPH and EQ from AEEM software performance is shown in the Figure-4. The vagueness and impreciseness in prediction of heterogeneous cross-project defect is prominently handled by proposed KNC-GWO and thus it performs better than k-means and fuzzy c means. While using real time dataset it is unavoidable about the presence of noisy and outlier instances, but if they are not considered during software defect prediction process it will create a negative impact during prediction process. Thus, fuzzy C means and K-means produce less result and their error rates are high as they fail to concentrate on imprecise instances such as noisy and outliers. Additionally, Grey wolf optimization enhances the process of kernel neutrosophic clustering to improve the centroid selection at the primary stage, so that the potential instances are considered at the initial stage to achieve accuracy with less time complexity.

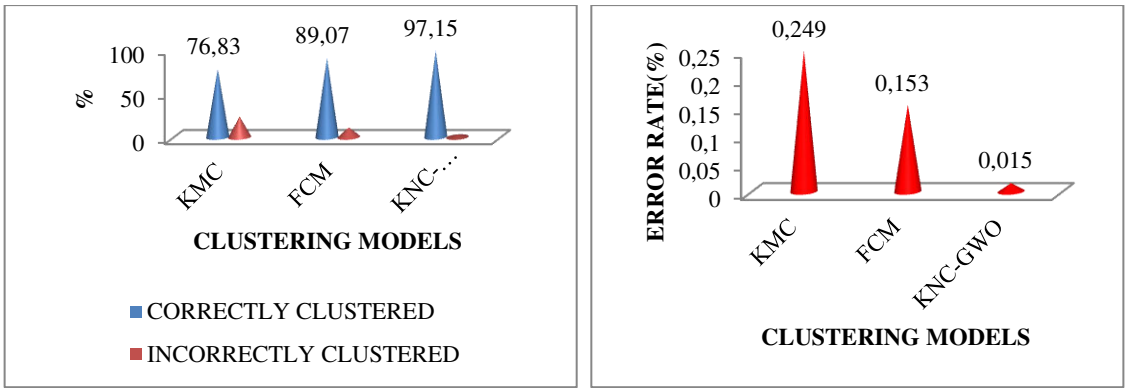

Figure-4. Performance comparison AEEM group dataset EQ and MORPH group dataset Velocity-1.4.

CONCLUSIONS

The main objective of this paper is to perform effective heterogeneous cross-project defect prediction using a behavioral inspired unsupervised learning model. There are two major issues involved in heterogeneous cross-project defect prediction they are two different datasets with various feature size has to be used and noisy as well as outlier has to be handled for better results. These problems are overcome by the developed model kernel neutrosophic clustering which inherits the property of representing each instance based on the membership degree of truthiness, falsity and indeterminacy. This ability intelligently categorizes the noisy and outlier instances which positively enhances the software defect prediction. The standard clustering algorithms doesn't consider vagueness and imprecise instances during clustering and they select the initial centroids in a random manner. The KNC is improvised by adapting the behavior of the grey wolf prey encircling strategy to produce the optimized clustering of buggy and clean datasets more positively and which is proved significantly from the obtained results.

REFERENCES

[1] He Peng, *et al*. 2015. An empirical study on software defect prediction with a simplified metric set. Information and Software Technology. 59: 170-190.

[2] Kim Dongsun, *et al*. 2013. Where should we fix this bug? a two-phase recommendation model. IEEE transactions on software Engineering 39.11: 1597-1610.

[3] Ryu Duksan, Okjoo Choi and Jongmoon Baik. 2016. Value-cognitive boosting with a support vector machine for cross-project defect prediction. Empirical Software Engineering 21.1: 43-71.

[4] Yin Xinglong, *et al*. 2020. Heterogeneous cross-project defect prediction with multiple source projects based on transfer learning. Mathematical Biosciences and Engineering 17.2 (2020): 1020-1040.

[5] Vashisht Rohit and Syed Afzal Murtaza Rizvi. 2020. Feature Extraction to Heterogeneous Cross-project Defect Prediction. 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE.

[6] Yang Xu, *et al*. 2020. Heterogeneous graph attention network for unsupervised multiple-target domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[7] Jahanshahi Hadi, Mucahit Cevik and Ayşe Başar. 2021. Moving from cross-project defect prediction to heterogeneous defect prediction: a partial replication study. arXiv preprint arXiv:2103.03490.

[8] Nam Jaechang, *et al*. 2017. Heterogeneous defect prediction. IEEE Transactions on Software Engineering 44.9: 874-896.

[9] Ni Chao, *et al.* 2017. FeSCH: a feature selection method using clusters of hybrid-data for cross-project defect prediction. 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). Vol. 1. IEEE.

[10] Menon, Aditya Krishna and Robert C. Williamson. 2003. Bipartite ranking: a risk-theoretic perspective. The Journal of Machine Learning Research 17.1 (2016): 6766-6867. F.Smarandache, A Unifying Field in Logics Neutrosophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability, third ed., American Research Press.

[11] Yang Miin-Shen and Hsu-Shen Tsai. 2008. A Gaussian kernel-based fuzzy c-means algorithm with a spatial bias correction. Pattern recognition letters 29.12: 1713-1725.

[12] Muro Cristian, *et al*. 2011. Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. Behavioural processes 88.3: 192-197.

[13] Jing Xiaoyuan, *et al*. 2015. Heterogeneous cross-company defect prediction by unified metric representation and CCA-based transfer learning. Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering. 496-507.

[14] D'Ambros Marco, Michele Lanza and Romain Robbes. 2010. An extensive comparison of bug prediction approaches. 2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010). IEEE.

[15] http://promise.site.uottawa.ca/SERepository/datasets-page.html