Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 4, Issue 4, July 2020
Part of the Proceedings of the 1st All India Paper writing Competition on Emerging Research - PaCER 2020

Exploring Principal Component Analysis in Defect Prediction: A Survey

Varsha G Palatse
Department of Computer Engineering, Government Polytechnic Pune, Maharashtra, India,
varshapalatse@gmail.com

Abstract: The performance of the defect prediction is solely based on the dataset which consist of software metrics. The software metrics or features sometimes very huge in number that makes the dataset complicated and it impacts the classifier or regressor efficiency. The dimension reduction technique is used to solve the problem of massive dimension of features. One of the most commonly used methods to deal with this issue is Principal Component Analysis (PCA). It is a statistical technique used for dimensionality reduction of the vast dataset in machine learning. Large number of research has been taken to predict the defective modules using principal component analysis. Its main function is to reduce the large number of features by extracting the uncorrelated features into groups. It helps to get simpler dataset, easy to handle and visualize, sometime in the cost of accuracy. In line with this, computation complexity of the predicators reduces and takes less time for execution. This survey paper is the exploration of the various studies conducted using principal component analysis for the defect prediction. The survey presented based on 28 studies from 2002 to 2020. It includes topics related to pre-release or post-release defects in software and machine equipment's related defects. The primary focus of this study is to examine the significance of principal component analysis, its contribution in defect prediction and to identify the expansion in this topic. The study will provide helpful outlines of this subject to on-topic scholars and experts.

Keywords: Defect Prediction Model; Principal Component Analysis; Feature Selection; Dimension Reduction; Machine Learning Algorithms; Eigenvector; Eigenvalue

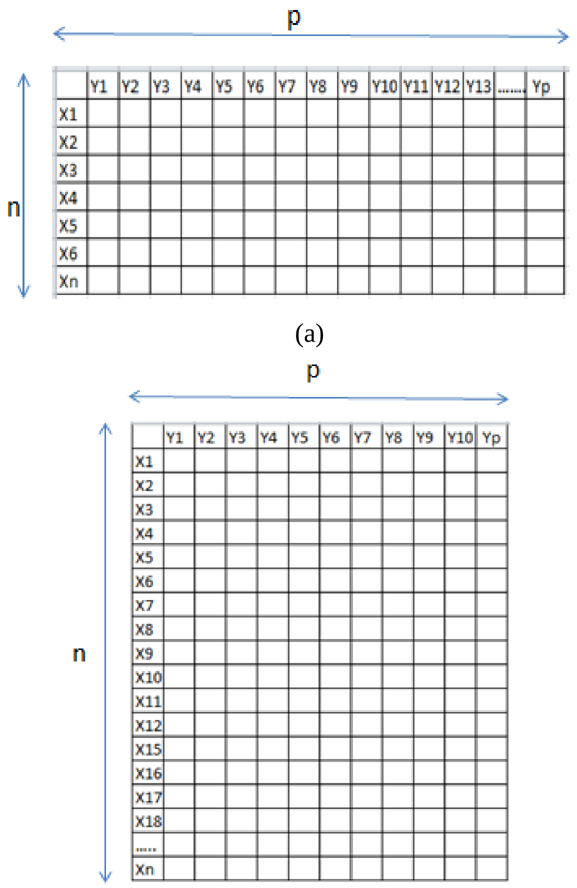
I. INTRODUCTION

The data has been the central point in the machine learning which contains huge number of features. In such situation, it is preferred to analyze the data and minimize the dimension of data to improve the efficiency and accuracy of the model. So, here dimension of data means the number of features or variables or attribute that are considered on each sample or record [3]. The problem "Big p Small n" [1] where dimension reduction is required because it has the number of features 'p' greater than the number of records 'n'. Statistically the number of records should be exponent to the count of features. But practically in some cases, the dimension of the features has increased.

The two ways of representation of data shown in Fig. 1, where X denotes records and Y denotes the features. Fig. 1a shows the problem of big p and small n where the number of features are large than number of records and Fig. 1b shows the statistical case where p<n. Fig. 1a also shows the popular problem called as the curse of dimensionality [1][4] which degrades the performance of the machine learning algorithm. The dimension reduction is one of the important steps in data pre-processing which eliminates the dimension of features. It identifies the relevant reduced set of dimensional representation without missing the information out of the original dataset [2]. Another significant point is that not all features are crucial for training the model. Before constructing any predictive model, we need to reduce the original data dimension without losing the important information [3]. This helps to keep the performance of model and handle the vast datasets.

The key motive of the dimension reduction is sum up as follows:

- The determination of the relevant reduced set of features those are useful for outcomes.
- Decrease the training time of the model.
- To preserve the important characteristics between features.
- To make visualization of high dimension more feasible and similar by mapping it to the lower dimension.
- Save the memory space.
- It checks the multicollinearity by removing the redundant features of same characteristics. For example, 'performed exercise in minutes' and 'fats burnt' are the features correlated with each other. They provide same kind of information of how much calories burnt. Hence, no need to store both the variable, one of them is sufficient [4].


p
n
Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Y11 Y12 Y13 .... Yp
X1 X2 X3 X4 X5 X6 Xn
p
n
Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Yp
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X15 X16 X17 X18 ..... Xn

(a)

(b)

Fig 1. Representation of Data; (a) "Big p Small n" problem; (b) Statistical point of view of data

Earlier studies [3][4] explained the various levels of the dimension reduction methods used in machine learning. These methods are the statistical based techniques used for analyzing the multivariate (multivariable) datasets. Methods such as principal component analysis (PCA) and factor analysis (FA) are very popular and linear dimension based methods. The resemblances and difference between these two methods explained in these studies [9][10]. Another method is fisher linear discriminant analysis(FLDA),which transform high dimensional data to low dimensional data by computing scattered features within and between the labels[11].Other methods that does not use covariance matrix are independent component analysis(ICA) - class of nonlinear PCA and projection pursuit[7][8]. Some other different methods are extensions and some works non-linearly. Fig. 2 shows various methods of dimension reduction , which has its own two kinds of dimensions : one is feature selection and another is feature transformation and further they are divided according to learning process[1][6].

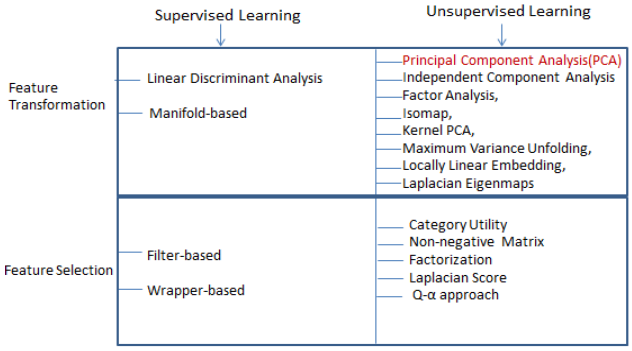


Fig 2. Different Dimension Reduction methods

Many studies carried out using the methods shown in Fig. 2 to improve the efficiency and accuracy of the models. To great extent principal component analysis method used for research to reduce the dimension of data. Principal component analysis is very simple to code and performs well in certain situation using some machine learning algorithm such as artificial neural network [12].It is linear dimension reduction method and uses the matrix manipulation of covariance. It retains the maximum characteristics of the original datasets even after reducing the high dimensional data to small dimensional data. It has been performed well particularly in post-release defect prediction [40] [41]. But in some cases principal components did not showed the effectiveness of the original datasets, because they were not original features. Also in some situation, the subset of features performed best than the smaller set of features [42] [43]. The defect prediction conducted for software during pre- release [23][24][34] or post-release of software[35][36] and in hardware related defect classification and diagnosis [26]. The prediction of fault-prone modules plays vital role in software quality assurance, which manages resources effectively and saves cost and time. It basically depends on the relevant subset of features extracted from the massive datasets [5] [13]. Hence, the study focuses on principal component analysis for defect prediction models as earlier no survey is carried out in this field as per information collected.

The purpose of survey is to explore various studies undergone through principal component analysis for defect prediction models. It involves the topics related to pre-release and post-release defects in software and some related to machine equipment's defect classification. Also the focus of this survey is to examine its contribution towards the models in terms of performance, accuracy and efficiency. This helps to identify its significance in this field and what more enhancements are required for better outcomes.

The paper is organized as; in Section 2 we provided the basics of PCA and important related concepts. Section 3 described the survey process and inclusion and exclusion factors for searching the relevant journals and conference papers. Section 4 represented the survey statistics. Section

5 elaborated the principal component analysis used in defect prediction. Section 6 concluded the survey study.

II. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is a mostly used statistical tool for analysis of data. It is simple and computes the statistics without prior knowledge of the form that can be used to draw the observations. It can handle the confusing, complex and massive data distribution and transform it into smaller and simpler datasets [18]. The drawn dataset consists of new small set of orthogonal and uncorrelated features which are known as principal components [19]. These components extract the relevant and maximum information from the original dataset. Hence, it is basically used for multivariate analysis in which it creates new uncorrelated variables (principal components) with maximum variance and minimum loss of information. The uncorrelated variables means correlation between any pair of variables should be 0. More specifically, if there is ten dimensional data, then there could be ten principal components. The first principal component extracts maximum information, then rest of the information by second component and likewise by remaining components [14] [15] as shown in Fig. 3.

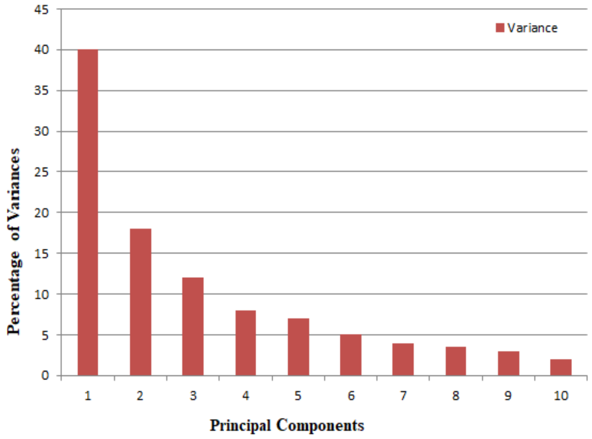
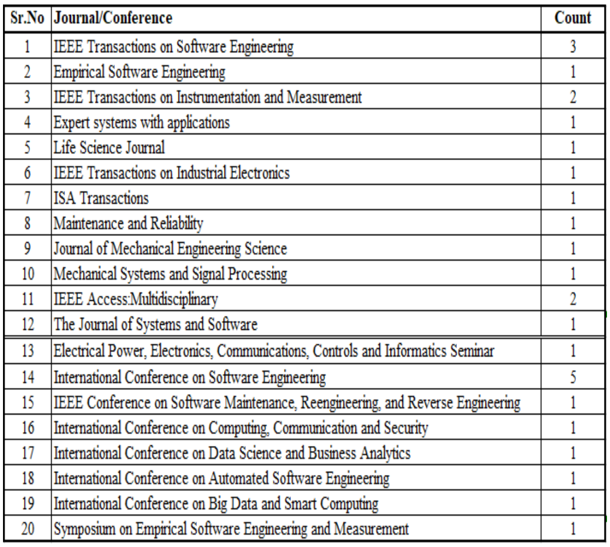


Fig 3. Principal Components with percentage of variances

PCA handles two types of variables – set of heterogeneous variables using multiple factor analysis [17] and qualitative variables by correspondence analysis [14] [16]. The principal component analysis is calculated on a square matrix – sum of squares and cross products (SSCP matrix) ,sums of squares and cross products from standardized data(Correlation matrix)[14] or scaled sums of squares and cross products (Covariance matrix)[1]. The eigenvectors and eigenvalues are the backbones for all the magic performed by the principal components. The eigenvectors of the covariance matrix are the directions of the axes where more variance is present means more information is available. And eigenvalues are the coefficients to eigenvectors which indicates how much variance acquired in each principal component [20] [21].Once the computation of eigenvectors done, arrange the eigenvalues in descending, and it gives the principal component in accordance of significance. After that we need to decide whether to use all components or remove the components whose eigenvalues are low. The thumb rule to select the number of components is 95% out of total variance. The resulting matrix is the feature vector which we decided to keep. This is the primary step of dimensionality reduction, where decided to keep only ‘p’ eigenvectors (components) out of n, then the reduced dataset have only p dimensions. The last step is to multiply the transpose of the feature vector by the transpose of the original dataset and get the final dataset [18]. For more detailed working of the PCA can be refer from these references.

III. SURVEY PROCEDURE

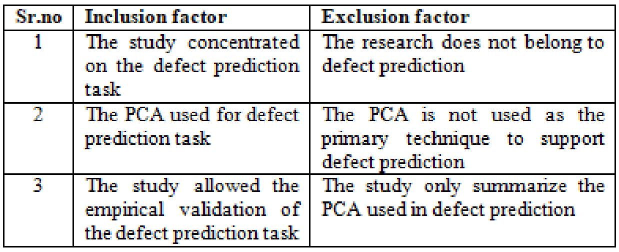
The process of survey initiated by searching through 12 conference papers and 16 journals in the defect prediction. All the journals and conference papers used are listed in the Table 1. To explore the sources on this topic, the search term used “principal component analysis “and “defect prediction”. During this look up process , the ‘ title ‘, ‘abstract ‘, ‘keyword’ used for searching. The searching scope restricted from year 2002 to 2020.



Sr.No	Journal/Conference	Count
1	IEEE Transactions on Software Engineering	3
2	Empirical Software Engineering	1
3	IEEE Transactions on Instrumentation and Measurement	2
4	Expert systems with applications	1
5	Life Science Journal	1
6	IEEE Transactions on Industrial Electronics	1
7	ISA Transactions	1
8	Maintenance and Reliability	1
9	Journal of Mechanical Engineering Science	1
10	Mechanical Systems and Signal Processing	1
11	IEEE Access:Multidisciplinary	2
12	The Journal of Systems and Software	1
13	Electrical Power, Electronics, Communications, Controls and Informatics Seminar	1
14	International Conference on Software Engineering	5
15	IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering	1
16	International Conference on Computing, Communication and Security	1
17	International Conference on Data Science and Business Analytics	1
18	International Conference on Automated Software Engineering	1
19	International Conference on Big Data and Smart Computing	1
20	Symposium on Empirical Software Engineering and Measurement	1

Table 1. Survey Statistics

After this look up process, we set the criteria to filter out the searched papers in order to get the relevant and applicable results that use the principal component analysis for defect prediction. The inclusion and exclusion factors used to find the results, shown in Table 2. For this survey, all three inclusion factors were considered in the study and the study was taken out if any one of the exclusion factors met.



Sr.no	Inclusion factor	Exclusion factor
1	The study concentrated on the defect prediction task	The research does not belong to defect prediction
2	The PCA used for defect prediction task	The PCA is not used as the primary technique to support defect prediction
3	The study allowed the empirical validation of the defect prediction task	The study only summarize the PCA used in defect prediction

Table 2. Selection Factors

IV. SURVEY DISCUSSION

On the basis of the search and filter activity, we selected 28 studies for survey of principal component analysis used in defect prediction models. With respect to publication, 16 papers from the journals and 12 papers from the conferences referred. The Fig. 4 represented the distribution of the selected papers in different years. It is analyzed that principal component analysis was mostly used during 2002 for predicting defects, after that decreased at constant value till 2011 and again increased the number of studies towards recent years.

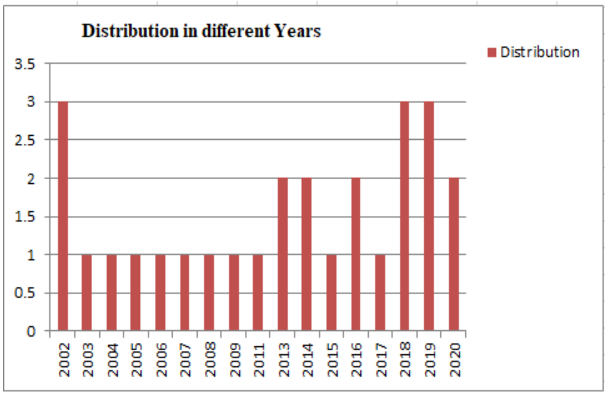
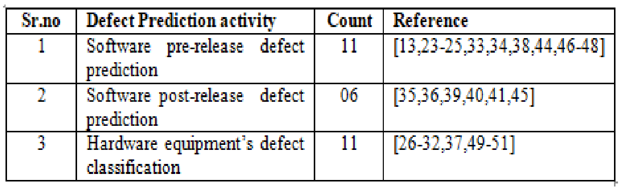


Fig 4. The distribution of the papers in different years

Principal component analysis is mostly used in the defect prediction models. The defect prediction is conducted during pre-release of software or post-release of software and for machine equipment's defect classification. Table 3 summarizes the principal component analysis used for defect prediction activity. We observed that there are 11 studies focusing on both the pre-release software defect prediction and hardware equipment defect detection. In addition, there are 06 studies used for software post-release defect prediction.



Sr.no	Defect Prediction activity	Count	Reference
1	Software pre-release defect prediction	11	[13,23-25,33,34,38,44,46-48]
2	Software post-release defect prediction	06	[35,36,39,40,41,45]
3	Hardware equipment's defect classification	11	[26-32,37,49-51]

Table 3. Principal Component Analysis used in the Defect Prediction Activity

V. PRINCIPAL COMPONENT ANALYSIS IN DEFECT PREDICTION

A. *Software Pre-release Defect Prediction*

The prediction of the fault before the release of software is very much important. It is a crucial step in software engineering. It has the capability to improve the software quality, helps in minimizing the testers and developers' effort on testing activity, save the cost on resources [22].

Denaro et al. proposed the study to investigate the presence of the classes of software for the existing fault-proneness models. These classes can solve problems of similar type, implemented using similar techniques in same environments and industries oriented. They used 9 orthogonal variables of around 96% of variance from the Apache 1.3 and 2.0 repositories for fault-proneness models using Logistic Regression. The outcome of principal component is lower but it required very less time for computation than the subset of features [23]. Briand et.al proposed study on object oriented data such as Jwriter, Xpose ,where they extracted 6 components with 76% of variance of data and revealed that the first component is less stronger than the other 5 components. They focused on the faulty classes' verification and explained that class ranking for fault-proneness is accurate. They supported that logistic regression is best suited for multivariate regression [24]. Neumann stated that some attributes showed common characteristics on risk of software. Rather using all attributes, only primary one should be used to constitute the cluster. In this study they presented the enhanced method for categorization of risk. It combined PCA with artificial neural network and encouraged the capability to detect high risk software. They clubbed the strong points of multivariate statistics, neural network and pattern recognition. PCA provided the normalized and orthogonal input data , which eliminates the multicollinearity issue. A neural network used to determine and classify risk in this study. They used the procedure called cross-normalization which supported to eliminate the datasets having disproportionate more number of high risk software modules [12]. Khoshgoftaar et.al conducted the empirical study on the real-time software system for fault prediction. They used six prediction models for comparison - 'CART-LS', 'CART-LAD', 'S-PLUS', 'CBR', 'ANN', and 'MLR' using original datasets and PCA for 4 large telecommunications system. The two-way ANOVA used with absolute average error and average relative error used as the response variables. From this study it is observed that PCA could not improve defect prediction accuracy , as original datasets and PCA gave same results , but PCA made the resultant models more strong[25].

Panichella et al. performed the empirical study and pointed that different classifiers captured different principal component and it varied from one project to other but Bayesian network captured the last component in all datasets. Also they analyzed that different methods assigned different fault proneness. They used the 10 Java projects on 6 different classifiers and identified different set of faulty prone classes [33]. Nagappan et.al implemented the study using relative churns which supported for increment in system defect density and also the performance of predictor increased. It distinguished the faulty and non-faulty modules. The principal components supported to reduce the correlated matrix [34]. Challagulla et. al investigated the capability of various machine learning algorithms on 4 datasets for prediction of defect. They evaluated the PCA on these algorithms and found that the PCA performed well for random forest, then ANN but worst for Navies Bayes. Also CFS outperformed than PCA [38].He et.al proposed the study for detection of class-imbalanced problem that affect the defect prediction performance. For this study they used ensemble multiboost depends on ripper classifier. They first identified the representative features and removed redundant ones by PCA. Then using synthetic and random sampling solved imbalanced problem. This study conducted using NASA datasets and equated with similar algorithms. This study outperformed the other techniques in evaluation measures [44].

Hadi et. al applied PCA to resolve the issue of correlated features and self-organizing map to get over of class imbalance on NASA datasets. Then compared the classification algorithms for optimum results. Random forest outperformed with highest accuracy of 96% than NB,SVM, J48 and IBk to detect defect in the dataset[46].Pak et. al performed data pre-processing on 17 datasets of PROMISE using their proposed approach PDT. They compared its result with PCA and feature subset selection using t-test and their approach worked better than PCA for defect prediction [47]. Menzies et.al elaborated that the effort and defect data hold local space that are different to the global space. It means whatever is useful for global scope might to useless for local scope. They demonstrated the local and global treatment using principal components [48].In this approach constructed local model that involved the training data clustering on WHERE algorithm and whatever outcomes obtained were classified using WHICH learning algorithm.

B. Software Post-release Defect Prediction

In Software Engineering, the maintenance phase plays very crucial role. It requires more efforts than any other phase in SDLC. It includes various activities such as correction, perfection, adaption and prevention. These activities are taking more time if the quality of code is poor, defective source code, undetected vulnerability etc. Hence maintainability of software is necessary to identify improvements areas and changes required for applications during development.

Nagappan et.al investigated that the ratio of software dependency and churn measures important for post-release defects for Windows Server 2003. The principal components contributed to estimate the faults occurred during field operation statistically. They carried out the study on logistic regression and computed correlation using Pearson and Spearman between forecasted fault probability and actual number of faults[35].Yamashita et.al investigated and analysed on the twelve code smells that the interactions between them affected maintenance and responsible for maintenance problems. They implemented on four Java projects with familiar smells. The smells were auto-detected in the pre maintenance variant of the systems. They recorded the problems they faced which factors related to them. PCA identified the ‘co-located code ‘smells. Also they discovered that smell interactions happened between coupled artefacts with negative impact as similar artefact co-location [36].

Nucci et.al performed empirical study on twenty-six open system to compare the accuracy of the prediction model with baseline methods by exploiting the process metrics- structural and semantic. The predictive model performed better and high complementary level on competitive methods. Also they conducted hybrid model for prediction more than 11 predictors, explored them by 5 competitive methods. The hybrid models have high accuracy as compared with 5 models. The PCA analysed that each predictor contributed differently and captured different components. Hence every model complementary to each other [39]. Zimmermann e.al proposed the system to detect the post-release faults for Windows Server 2003, they evaluated that network measures predicted the number of defects more than complexity metrics. Here they conducted experiment on three models using principal components and reported the 99% correlations [40]. Nagappan et.al proposed empirical study on 5 Microsoft systems for post-release faults. They discovered that code complexity metrics related to defect prone modules. They created the regression model and showed that the defects predicted accurately using new sets of features extracted by PCA [41]. Kumar et. al applied three AI techniques on two cases quality evaluation system and user interface system for predicting maintainability. In their approach, principal component analysis used to extract variables that are not related to each other and rough set analysis used to capture unique features that decreased the precision degree. These features more visible to mine factual data. They concluded that their approach outperformed than existing one and small set of features contributed for prediction with rich accuracy [45].

C. Hardware Equipment Defect Prediction

To prevent the sudden shut down of machinery and to avoid the catastrophic damages of the machines, it is required detect the failures of important parts of the machines. Therefore it is necessary to monitor the conditions of the parts to know details of defects severity ahead of the serious consequences.

Malhi et.al proposed the system to detect the defect severity for bearings using supervised and unsupervised classification approaches. In supervised approach PCA selected appropriate features and given to feed forward neural networks and radial basis function. This investigated the defect classification. For unsupervised training, the most dominant features were identified by computing the vibrations of defected bearings without the previous information of defect. The identified features given to learning scheme to class the defected bearing using the defect size. They performed this study by using three dissimilar bearing fault configurations and showed that the accuracy of the classification improved using PCA. This technique presented was general for any problem [26]. Widodo et.al proposed fault detection using relevance vector machine and support vector machine They detected fault of slow speed bearings using acoustic emission and vibration signal. The component analysis performed for feature extraction and high dimension reduction of the dataset. It showed that RVM with feature extraction method performed very well as compared with SVM [27]. Seryasat et.al proposed system that diagnosed the faults in bearing to avoid its malfunctioning at the time of operation. They used the 12 features from frequency and time domain and reduced them to 6 using principal components. As features reduced, still accuracy of average diagnosis not decreased. These features contributed to multiclass support vector machine for classification performance [28].

You et.al proposed the creative approach for welding monitoring and welded defect detection in that the costly sensor with complex structure replaced with cheap and simple sensor. They performed using the multivariate analysis and feed-forward NN and support vector machine. For detecting welded defect, they used pattern recognition where three techniques used to acquire welded images. The principal components of spectrometer and photodiode improved prediction and accuracy of defect classifiers [29]. Saidi et.al proposed the novel approach for diagnosis of bearings, used the high order features and support vector machine model. The non-linear features by high order range were used to analyze vibration signals. The bi-spectrum vibration used as feature vectors for differentiating faults of bearings. These feature vectors were the principal components used to improve the performance of algorithms [30].Also they performed the 10 fold cross-validation and obtained the minimal parameters for classification. Zuber et.al implemented the fault detection approach for bearings using vibration features as input to the artificial neural network. The vibration features resulted from principal components analysis and they revealed that these features were capable to identify faults and enhanced the performance of ANN [31]. Zair et.al proposed the combined approach using three methods together and diagnosed the multi-class fault of roll bearings. The principal components were feature vectors constructed by entropy based fuzzy of empirical mode decomposition. These vectors used as input to SOM for detection and classification of fault. This approach proved that, it recognised different kinds of defects of bearings [32]. Aouabdi et.al proposed an approach that identified defects of gear tooth developed on MCSA using MSE and classifier. They used 4 statistics on MSE for defect classification and multivariate approach. PCA-MSE inspected the complex down sampled signal biased by artifactual components. The approach showed that this method capable to detect gear tooth pitting defect in signals[37].

Yao et. al proposed the approach to decrease the workload of rail maintenance under high standards which in turn reduced the labour cost for the predicted sections that don’t have any defects with 100% recall rate[49]. Wang et.al proposed the algorithm to isolate the multiple source partial discharge signals which counted the clustering degree and one parameter added based on original variables. PCA reduced the feature space from 12-D to 2-D space ,still this algorithm separated the multiple PD sources in effect. This PD detection is an efficient way to predict and diagnose the power equipment’s insulation condition [50].Chang et. al proposed the approach to accurately predict the SMT solder joint using classification algorithms. It combined PCA with clustering based on similarity and density. This approach reduced complex computation by using PCA and provided the accuracy. The deep neural network predicted successfully failed components before inspection of AOI machine. This algorithm improved false judgement of AOI [51].

VI. CONCLUSIONS

In this paper, a survey of principal component analysis in defect prediction is presented based on 28 selected publications from 2002 to 2020. From the survey, we observed that principal component analysis is widely used in various defect prediction activities in increasing trend. However, it is highly used during 2002’s and in recent years now. These activities include software pre-release, software maintainability and hardware equipment defect prediction. Among these activities, software early defect prediction and hardware related defect prediction widely used principal component analysis.

ACKNOWLEDGMENTS

I would like to acknowledge the anonymous reviewers for the valuable comments and suggestions that improved the quality of this paper. Moreover, this study is carried out independent of the associated institute.

REFERENCES

[1] P. Cunningham, “Dimension reduction,” Cogn. Technol., vol. 1, pp. 91–112, 2008, doi: 10.1201/b18358-4.

[2] N. Kambhatla and T. K. Leen, “Dimension Reduction by Local Principal Component Analysis,” Neural Comput., vol. 9, no. 7, pp. 1493–1516, 1997, doi:10.1162/neco.1997.9.7.1493.

[3] I. K. Fodor, “A survey of dimension reduction techniques,” Library (Lond)., vol. 18, no. 1, pp. 1–18, 2002, doi: 10.2172/15002155.

[4] M. Carreira-Perpinán, “A review of dimension reduction techniques,” Dep. Comput. Sci. Univ. Sheffield. Tech. Rep. CS-96-09, pp. 1–69, 1997.

[5] J. Li et al., “Feature selection: A data perspective,” ACM Comput. Surv., vol. 50, no. 6, 2017, doi: 10.1145/3136625.

[6] C. Bartenhagen, H. U. Klein, C. Ruckert, X. Jiang, and M. Dugas, “Comparative study of unsupervised dimension reduction

techniques for the visualization of microarray gene expression data," BMC Bioinformatics, vol. 11, 2010, doi: 10.1186/1471-2105-11-567.

[7] A. Hyvarinen, "Survey on Independent Component Analysis," Neural computing surveys vol. 3, no. 2, pp. 54–67, 1999.

[8] P. Comon. Independent Component Analysis, a new concept?. Signal Processing, Elsevier, 1994, 36, pp.287-314. 10.1016/0165-1684(94)90029-9. hal-00417283.

[9] N. E. Benton and M. Neil, "A critique of software defect prediction models," IEEE Trans. Softw. Eng., vol. 25, no. 5, pp. 675–689, 1999, doi: 10.1109/32.815326.

[10] P. M. Jain and V.K. Shandliya, "A survey paper on comparative study between Principal Component Analysis (PCA) and Exploratory Factor Analysis (EFA)," Int. J. Comput. Sci. Appl., vol. 6, no. 2, pp. 373–375, 2013.

[11] A. Kalsoom, M. Maqsood, M. A. Ghazanfar, F. Aadil, and S. Rho, A dimensionality reduction-based efficient software fault prediction using Fisher linear discriminant analysis (FLDA), vol. 74, no. 9. 2018.

[12] D. E. Neumann, "An enhanced neural network technique for software risk analysis," IEEE Trans. Softw. Eng., vol. 28, no. 9, pp. 904–912, 2002, doi: 10.1109/TSE.2002.1033229.

[13] V. Palatse, "Feature selection techniques for software defect prediction: A literature review," Test Eng. Manag., vol. 83, no. 2245, pp. 2245–2253, 2020.

[14] H.Abdi and L. J. Williams, "Principal Component Analysis," Wiley interdisciplinary reviews: computational statistics, pp. 1–10, 2010.

[15] J Lever, M Krzywinski, N Altman," POINTS OF SIGNIFICANCE Principal component analysis," © Nature America, Inc., part of Springer Nature. – 2017

[16] G. Der and B. Everitt, "Correspondence Analysis," Handb. Stat. Anal. Using SAS, Second Ed., 2001, doi: 10.1201/9781420057553.ch16.

[17] H. Abdi, L. J. Williams, and D. Valentin, "Multiple factor analysis: Principal component analysis for multitable and multiblock data sets," Wiley Interdiscip. Rev. Comput. Stat., vol. 5, no. 2, pp. 149–179, 2013, doi: 10.1002/wics.1246.

[18] J.Shlens,"A tutorial on principal component analysis," arXiv preprint arXiv:1404.1100(2014)

[19] B. S. Everitt and J. E. Jackson, "A User's Guide to Principal Components.," Biometrics, vol. 48, no. 3, p. 974, 1992, doi: 10.2307/2532367.

[20] L.Smith "A tutorial on Principal Components Analysis , "Department of Computer Science, University of Otago, Artif. Intell., pp. 1–4, 2004.

[21] H.Abdi , "The Eigen-decomposition: Eigenvalues and eigenvectors." Encyclopedia of measurement and statistics (2007): 304-308.

[22] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A Systematic Literature Review on Fault Prediction Performance in Software Engineering," pp. 1–31, 2011.

[23] P. Milano, "An Empirical Evaluation of Fault-Proneness Models Giovanni D e n a r o," pp. 241–251.

[24] L. C. Briand, "W. L..Melo, and J. Wust. "Assessing the applicability of fault-proneness models across object-oriented software projects." IEEE transactions on Software Engineering 28.7 (2002): 706-720.

[25] T. M. Khoshgoftaar and N. Seliya, "Fault prediction modeling for software quality estimation: Comparing commonly used techniques," Empir. Softw. Eng., vol. 8, no. 3, pp. 255–283, 2003, doi: 10.1023/A:1024424811345.

[26] A. Malhi and R. X. Gao, "PCA-based feature selection scheme for machine defect classification," IEEE Trans. Instrum. Meas., vol. 53, no. 6, pp. 1517–1525, 2004, doi: 10.1109/TIM.2004.834070.

[27] A. Widodo, E. Y. Kim, J-D Son , B-S Yang , A. C.C. Tan , D-S Gu , B-K Choi , J. Mathew , "Fault diagnosis of low speed bearing based on relevance vector machine and support vector machine," Expert Syst. Appl., vol. 36, no. 3 PART 2, pp. 7252–7261, 2009, doi: 10.1016/j.eswa.2008.09.033.

[28] O.R. Seryasat , H. G. Zadeh , M. Ghane , Z. Abooalizadeh , A. Taherkhani , F. Maleki , "Fault Diagnosis of Ball-bearings Using Principal Component Analysis and Support-Vector Machine," Emerg. Infect. Dis., vol. 4, no. 1, pp. 1–7, 2013, doi: 10.1016/S0304-4017(96)01152-1.

[29] D. You, X. Gao, and S. Katayama, "WPD-PCA-based laser welding process monitoring and defects diagnosis by using FNN and SVM," IEEE Trans. Ind. Electron., vol. 62, no. 1, pp. 628–636, 2015, doi: 10.1109/TIE.2014.2319216.

[30] L. Saidi, J. Ben Ali, and F. Fnaiech, "Application of higher order spectral features and support vector machines for bearing faults classification," ISA Trans., vol. 54, pp. 193–206, 2015, doi: 10.1016/j.isatra.2014.08.007.

[31] N. Zuber and R. Bajrić, "Application of artificial neural networks and principal component analysis on vibration signals for automated fault classification of roller element bearings," Eksploat. i Niezawodn. - Maint. Reliab., vol. 18, no. 2, pp. 299–306, 2016, doi: 10.17531/ein.2016.2.19.

[32] M. Zair, C. Rahmoune, and D. Benazzouz, "Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network," Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci., vol. 233, no. 9, pp. 3317–3328, 2019, doi: 10.1177/0954406218805510.

[33] A. Panichella, R. Oliveto, and A. De Lucia, "Cross-project defect prediction models: L'Union fait la force," 2014 Softw. Evol. Week - IEEE Conf. Softw. Maintenance, Reengineering, Reverse Eng. CSMR-WCRE 2014 - Proc., pp. 164–173, 2014, doi: 10.1109/CSMR-WCRE.2014.6747166.

[34] N. Nagappan and T. Ball, "Use of relative code churn measures to predict system defect density," Proc. - 27th Int. Conf. Softw. Eng. ICSE05, pp. 284–292, 2005, doi: 10.1145/1062455.1062514.

[35] N.Nachiappan and T.Ball,"Using software dependencies and churn metrics to predict field failures: An empirical case study". First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007).IEEE 2007.

[36] A. Yamashita and L. Moonen, "Exploring the impact of inter-smell relations on software maintainability: An empirical study," Proc. - Int. Conf. Softw. Eng., pp. 682–691, 2013, doi: 10.1109/ICSE.2013.6606614.

[37] S. Aouabdi, M. Taibi, S. Bouras, and N. Boutasseta, "Using multi-scale entropy and principal component analysis to monitor gears degradation via the motor current signature analysis," Mech. Syst. Signal Process., vol. 90, pp. 298–316, 2017, doi: 10.1016/j.ymssp.2016.12.027.

[38] G. P. Bhandari and R. Gupta, "Machine learning based software fault prediction utilizing source code metrics," Proc. 2018 IEEE 3rd Int. Conf. Comput. Commun. Secur. ICCCS 2018, pp. 40–45, 2018, doi: 10.1109/CCCS.2018.8586805.

[39] D. Di Nucci, F. Palomba, G. De Rosa, G. Bavota, R. Oliveto, and A. De Lucia, "A Developer Centered Bug Prediction Model," IEEE Trans. Softw. Eng., vol. 44, no. 1, pp. 5–24, 2018, doi: 10.1109/TSE.2017.2659747.

[40] T. Zimmermann and N. Nagappan, "Predicting defects using network analysis on dependency graphs," Proc. - Int. Conf. Softw. Eng., pp. 531–540, 2008, doi: 10.1145/1368088.1368161.

[41] N. Nagappan, T. Ball, and A. Zeller, "Mining metrics to predict component failures," Proc. - Int. Conf. Softw. Eng., vol. 2006, pp. 452–461, 2006, doi: 10.1145/1134285.1134349.

[42] X. Yang, K. Tang, and X. Yao, "A learning-to-rank approach to software defect prediction," IEEE Trans. Reliab., vol. 64, no. 1, pp. 234–246, 2015, doi: 10.1109/TR.2014.2370891.

[43] M. D'Ambros, M. Lanza, and R. Robbes, "Evaluating defect prediction approaches: A benchmark and an extensive comparison," Empir. Softw. Eng., vol. 17, no. 4–5, pp. 531–577, 2012, doi: 10.1007/s10664-011-9173-9.

[44] H. He , X. Zhang, Q. Wang, J. Ren , J.Liu, X..Zhao And Y. Cheng, "Ensemble MultiBoost Based on RIPPER Classifier for Prediction of Imbalanced Software Defect Data," IEEE Access, vol. 7, pp. 110333–110343, 2019, doi: 10.1109/access.2019.2934128.

[45] L. Kumar and S. K. Rath, "Hybrid functional link artificial neural network approach for predicting maintainability of object-oriented software," J. Syst. Softw., vol. 121, pp. 170–190, 2016, doi: 10.1016/j.jss.2016.01.003.

[46] N. T. Hadi and S. Rochimah, "Enhancing Software Defect Prediction Using Principle Component Analysis and Self-Organizing Map," 2018 Electr. Power, Electron. Commun. Control. Informatics Semin. EECCIS 2018, pp. 320–325, 2018, doi: 10.1109/EECCIS.2018.8692889.

[47] C. M. Pak, T. T. Wang, and X. H. Su, "Software Defect Prediction Using Propositionalization Based Data Preprocessing: An Empirical Study," Proc. - 2nd Int. Conf. Data Sci. Bus. Anal. ICDSBA 2018, pp. 71–77, 2018, doi: 10.1109/ICDSBA.2018.00021.

[48] T. Menzies, A. Butcher, A. Marcus, T. Zimmermann, and D. Cok, "Local vs. global models for effort estimation and defect prediction," 2011 26th IEEE/ACM Int. Conf. Autom. Softw. Eng. ASE 2011, Proc., pp. 343–351, 2011, doi: 10.1109/ASE.2011.6100072.

[49] N.Yao , J. Yuejun, and T. Kai. "Rail Weld Defect Prediction and Related Condition-Based Maintenance." IEEE Access 8 (2020): 103746-103758.

[50] Y. B. Wang, D. G. Chang, S. R. Qin, Y. H. Fan, H. B. Mu, and G. J. Zhang, "Separating multi-source partial discharge signals using linear prediction analysis and isolation forest algorithm," IEEE Trans. Instrum. Meas., vol. 69, no. 6, pp. 2734–2742, 2020, doi: 10.1109/TIM.2019.2926688.

[51] Y. M. Chang, C. C. Wei, J. Chen, and P. Hsieh, "An Implementation of Health Prediction in SMT Solder Joint via Machine Learning," 2019 IEEE Int. Conf. Big Data Smart Comput. BigComp 2019 - Proc., pp. 12–15, 2019, doi: 10.1109/BIGCOMP.2019.8679428.