Improved Software Defect Prevention using Transfer Learning

P. Sampath Kumar
Asst Professor Dept. of CSE
P.S.G College of Tech
Coimbatore, India

R. Venkatesan, PhD
Professor, Dept. of CSE
P.S.G College of Tech
Coimbatore, India

ABSTRACT

This paper presents a highly effective technique to handle data insufficiency issue in software defect prevention using machine learning techniques. Extracting knowledge using data mining techniques in software engineering is a difficult task as the data available from software projects for research is not only less but also outdated. Generally Software engineering activities like defect prediction, effort estimation etc., were done on data available from open source datasets which is less in volume.

All researchers and data scientists tend to agree on one common thing i.e. they always need more quality data to produce accurate results. When the data used to construct models is lesser than essential quantity, the results predicted will be inaccurate and unstable. In this paper, transfer learning technique has been employed to tackle this data insufficiency issue using techniques, by transferring knowledge from related similar task, where sufficient data is available and this extracted knowledge is made use in the pursuing task to create more accurate prediction. From the experimental results, it is evident that transfer learning technique employed show considerable improvement in defect prevention even when the data available for that problem is limited.

General Terms

Software Defect Prediction, Data Insufficiency

Keywords

Transfer Learning, Machine Learning, Artificial Neural Networks

1. INTRODUCTION

The need for more data to extract knowledge to do accurate prediction has risen over the years in software project management activities like software effort estimation, software defect prediction etc. Software defects released along with the product can make the applications unreliable, make the customer incur huge loss in terms of money and time and erode the confidence of the customer [1].

In order to assure the quality and reliability of released software products, software defect prediction needs to be accurate. At present, Artificial Intelligence techniques are employed to predict whether the software module is defective or not from the previous data history. When the defective module is identified through the defect prediction process, then skilled personnel [2] can be employed to concentrate on high risk modules efficiently for early defect detection which will improve the product quality with less time and cost[3].

To get accurate results in learning models, data should be large enough and should be of good quality. The previous software project engineering data available in industries are limited and it is also difficult to find similar relevant projects from other firms, as they do not release or expose the software project data to outside world for analysis and research. In order to encounter these data scarcity issues, suitable techniques need to be adopted to increase the accuracy of defect prediction in the early part of software lifecycle.

The accuracy of the classification model gets affected when the data available is less. The implications of data insufficiency and possible approaches to solve them are shown in Figure 1. Lesser data availability for model development leads to difficulty in optimization and lack of generalization issues. Every technique identified to solve these data insufficiency issues has specific methods available and these specific methods can be applied to overcome the data insufficiency issue and improve the defect prediction accuracy.

In this paper, the data insufficiency issue is overcome using a technique known as Transfer Learning (Homogeneous method). Transfer learning(TF) is a technique employed in transferring knowledge between two separate but related problems which can be used to improve the performance of the problem under study. Usually this method is adapted in computer vision domain where they train a convolutional neural network or artificial neural network in a related project where more images are available. The patterns learnt from this project are used in constructing a model and fine tuning it for the project with fewer images. In this work transfer learning is employed to acquire knowledge to build a software defect model from a task where data is large and then use that knowledge to tune the target model for a task that has lesser data.

2. THEORETICAL BACKGROUND

Data should be sufficiently large to make an accurate prediction or classification from any machine learning model. Transfer learning can be applied to a project with less data when a related project is found and useful structure and knowledge can be extracted from that related project that can be applied in the current project.

2.1 Transfer Learning:

Transfer learning technique mitigates the notion that the training and test data used in learning model should be independent and identically distributed (i.i.d) [4]. In order to improve the prediction accuracy of task T_t with data D_t in the target domain whose size is less, transfer learning is applied which extracts knowledge from source task with source data(Ds). Transfer learning improves the P(Yt/Xt) conditional probability from the knowledge obtained from Ds where Yt=Target label and Xt=Target feature vector [5].

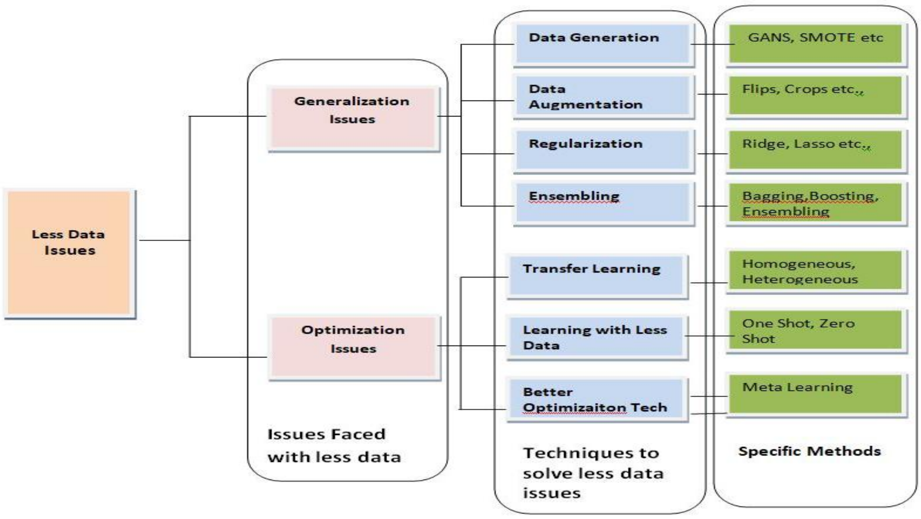

Figure 1: Data Insufficiency issues and Techniques to handle them

Transfer learning can be of homogeneous or heterogeneous types. In homogeneous transfer learning, the feature space (input feature vectors and label) is the same ($X_t = X_s$ and $Y_t = Y_s$) for both source and target task. In heterogeneous transfer learning, the feature and label spaces of source and target domain are different (($X_t \neq X_s$) and/*or* ($Y_t \neq Y_s$)).

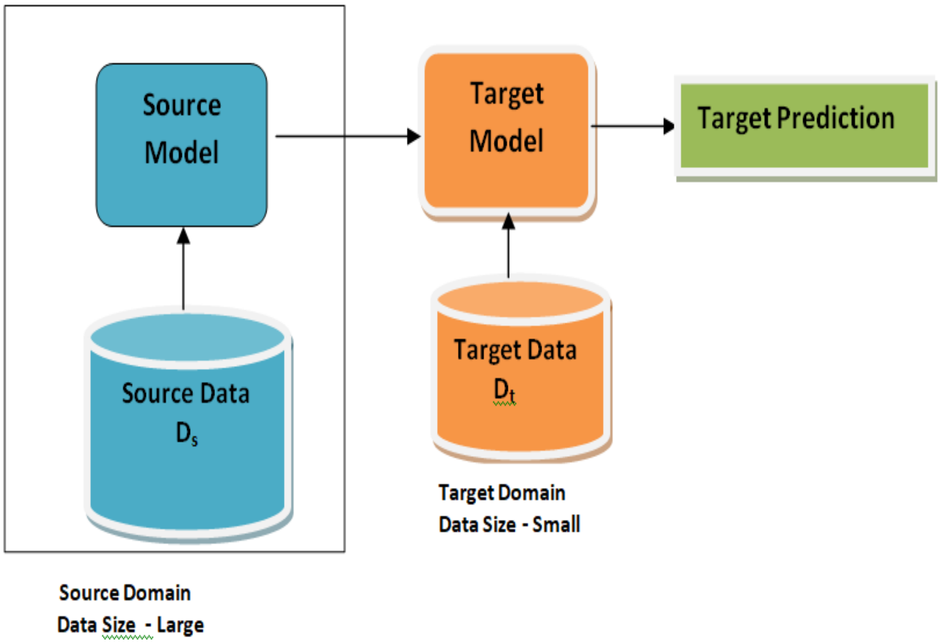

Figure 2: Transfer Learning

In this work homogeneous transfer learning has been employed where the feature and label spaces in the two related projects are the same as shown in Figure 2. In software engineering, especially in software defect prediction problem, it is difficult to find larger datasets fomodel is used in transfer learning and the MLP Model obtained using the source data is saved and retrieved for the target domain.

3. RELATED WORK

In the recent past, lot of research has been made, to increase the accuracy of early defective module prediction so that there will be high improvement in final quality of product delivered. Software defect prediction needs to be as accurate as possible, as software defects and faults delivered to the customer will be expensive and detrimental to the customer satisfaction. Jones et al [6] details about detecting and rectifying expenses of defects in software project activities to be really high.

Wahono reported that software defect prediction in the last four decades have focused on methods to identify whether any software component will be defect prone or not using static attributes[7]. The classification methods cover various machine learning (ML) algorithms like ANN, Logistic Regression, Naïve Bayes and SVM. In order to improve accuracy further researchers resorted to ensemble methods like boosting, bagging and stacking [8].

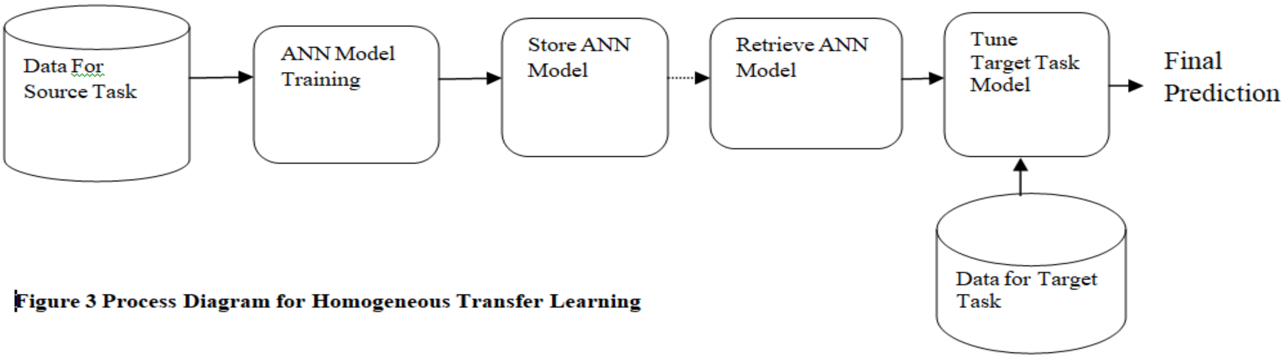

Figure 3 Process Diagram for Homogeneous Transfer Learning

Defect prediction works well, if models are trained with a sufficiently large amount of data and applied to a single software project. But in reality, large training data is not available in public for research, as organizations do not expose all their software engineering data or they do not maintain long project data history. In order to tackle this issue, either additional data need to be generated or the knowledge obtained from the previous task has to be transferred to another related task. Zimmerman found that the problems/tasks should be similar and in the same domain to extract knowledge and use that to improve prediction accuracy. If the tasks chosen are from different domain, the accuracy will be poor [9].

Researchers have used the Software Defect data from NASA Metrics data program (MDP) extensively for defect classification using various machine learning algorithms [10]. The NASA MDP data set needs to be preprocessed to yield accurate classification. The data needs to be cleansed and the appropriate independent variables (feature subset) need to be selected to improve the performance accuracy [11].

Transfer learning becomes an inevitable technique in machine learning to deal with the issue of insufficient training data, when another relatable and similar task/project can be identified with sufficient data. This technique tries to transfer the knowledge by softening the hypothesis that the data used for training and testing must be i.i.d (independent and identically distributed) [12]. Homogeneous Transfer learning is used when the feature space and the label space are the same in both the domains [13].

Having sufficient historical software defect prediction data can be a challenge, especially for newer projects and legacy projects. In order to improve accuracy cross project defect prediction is employed which uses transfer learning to gain knowledge from related and similar software engineering projects [14].

4. PROPOSED WORK:

In this work, an effective technique is proposed to improve the software defect prevention by using methods which will overcome the data insufficiency issue. For this work, transfer learning has been employed on NASA MDP datasets to illustrate how this technique can be used to overcome the data size issue and improve the defect prediction accuracy.

4.1 Data Pre-processing:

NASA MDP defect dataset contains irregularities in the form of missing values and irrelevant features. The missing values are replaced by mean imputation where mean of the columns replaces the missing value. After that, the most relevant features are found by assigning a value to each independent variable which is known as feature importance which indicates the relative importance of each feature in making the classification. The filtering of irrelevant independent variables will improve the accuracy of the model. By selecting using feature importance criteria, the relevant features(independent variables) is filtered from 22 independent variables to 10 relevant variables [15][16].

4.2 Transfer Learning

Transfer learning technique is a useful technique when the data available is less in quantity for constructing machine learning and deep learning models.

To overcome the data insufficiency issue, the structural knowledge or pattern is extracted from a relative task or a project where data is sufficiently large and then transferred to the task with scarce data. Multi Layer perceptron has been selected as the model in this transfer learning technique experiment carried out to deal with data scarcity. Transfer learning greatly helps in not only decreasing the training time for a neural network but also produces lower generalization error.

The process diagram for the homogeneous transfer learning is shown in Figure 3. Multi layer perceptron (MLP) model is first trained on the source task which is identical to the task that is being solved where more data is available. The trained MLP (neural network) model in its entirety is stored on to a file. Here homogeneous transfer learning has been used, where the feature space and label space are similar, the saved model is retrieved and the retrieved neural network model is used as the starting point for the target task model learning. The target task is then fine tuned using the training data which is used for predicting the final response for the target task

Table 1: Evaluation Metrics for NASA MDP Datasets with and without Transfer Learning

Source Dataset (Instances)	Target Dataset	Accuracy		Precision		Recall		F-Measure		Specificity	
		With TF	Without TF	With TF	Without TF	With TF	Without TF	With TF	Without TF	With TF	Without TF
KC1 (2109)	CM1(498)	.94	.90	.88	.82	.94	.89	.91	.86	.84	.79

PC1 (1109)	CM1 (498)	.91	.89	.88	.86	.92	.89	.86	.86	.82	.79
JM1 (10885)	KC1 (2109)	.88	.84	.80	.80	.84	.83	.80	.75	.76	.74
JM1 (10885)	CM1 (498)	.90	.89	.86	.85	.90	.89	.87	.86	.82	.79

5. RESULTS AND DISCUSSION

This section reports the results of the experiments conducted with the NASA MDP data using transfer learning and does a comparison of the defect prediction results obtained with transfer learning and without transfer learning. In this experiment, transfer learning has been employed using Multi layer perceptron using the NASA MDP datasets. In each scenario Multilayer perceptron (MLP) model with weights learnt from the source data is stored. The stored model is retrieved and used as initialization for the target task defect classification. The defect classification metrics [17] are measured and then compared with the metrics of the experiment done without transfer learning. The results of both the experiments are tabulated in Table 1.

From Table 1 it is observed that there is improvement in these metrics (accuracy, precision, recall, F-measure and specificity) in the target task when transfer learning is employed. The bar chart (Figure 4) compares the classification metrics accuracy, precision, recall, F-Measure and specificity in four transfer learning tasks between datasets obtained from NASA MDP data repository. The improvement in the classification metric values is due to better initialization points used in the target model predictions. The better initialization point depends not only on the quantity of data in the source model but also on the quality of the data used in the source model. The quality of the NASA MDP data is improved by applying extensive data pre-processing techniques.

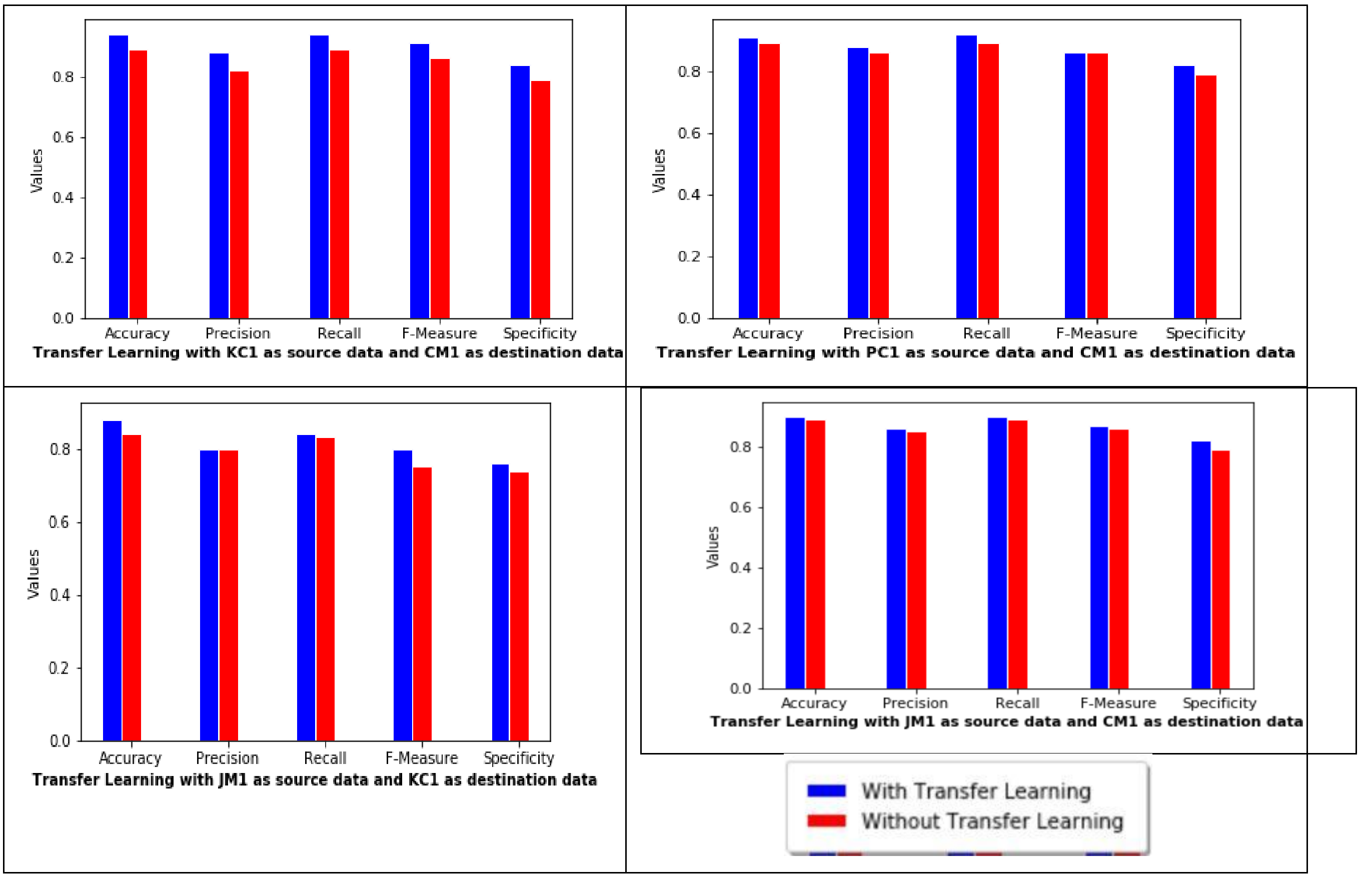

Figure 4: Bar Chart for Four Experimental Tasks with and without Transfer Learning

6. CONCLUSION

Software engineering datasets (especially defect prevention data) are not largely available to build accurate machine learning models, as industries do not expose their projects' data or have large enough project history. In order to deal with this issue, techniques to overcome data scarcity need to be applied to build a near accurate model for defect prevention. In this paper transfer learning technique is employed successfully to deal with the issue of data insufficiency in software defect prevention using NASA MDP datasets which are available for public. From the results obtained for the defect classification metrics, it is clear that the usage of transfer learning technique has helped to overcome the data insufficiency issue.

This improvement in the software defect evaluation metrics can be applied in real world software industries also, if similar and relevant projects with larger datasets can be identified and then used as source model. Heterogeneous transfer learning can be applied when the source and the target projects participating in transfer learning differ in feature spaces. Using this work, it is shown that transfer learning can be successfully applied to areas in software engineering using numerical data when transfer technique was mostly employed for computer vision tasks.

7. ACKNOWLEDGMENTS

The authors would like to thank the Principal and Management of P.S.G College of Technology, Coimbatore India for all their support rendered in carrying out this project successfully.

8. REFERENCES

[1] N. E. Fenton and S. L. Pfleeger, "Software metrics: a rigorous and practical approach", PWS Publishing Co., (1998).

[2] W. Zheng, S. Mo, X. Jin, Y. Qu, Z. Xie, and J. Shuai, "Software defect prediction model based on improved deep forest and AutoEncoder by forest," *Proc. Int. Conf. Softw. Eng. Knowl. Eng. SEKE*, vol. 2019-July, no. 3, pp. 419–424, 2019.

[3] R. Li, L. Zhou, S. Zhang, H. Liu, X. Huang, and Z. Sun, "Software Defect Prediction Based on Ensemble Learning," *ACM Int. Conf. Proceeding Ser.*, pp. 1–6, 2019.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[5] Sebastian Ruder:Transfer Learning - Machine Learning's Next Frontier 2017 [Online] https://ruder.io/transfer-learning/ [Accessed: 05- May- 2020].

[6] Jones, C., & Bonsignour, O. (2012). The Economics of Software Quality. Pearson Education, Inc.

[7] R. S. Wahono, "A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks," *J. Softw. Eng.*, vol. 1, no. 1, pp. 1–16, 2015.

[8] J. Petrić, D. Bowes, T. Hall, B. Christianson, and N. Baddoo, "Building an Ensemble for Software Defect Prediction Based on Diversity Selection," *Int. Symp. Empir. Softw. Eng. Meas.*, vol. 08-09-September-2016, 2016.

[9] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, "Cross-project defect prediction," p. 91, 2009.

[10] M. Shepperd, Q. Song, Z. Sun, and C. Mair, "Data quality: Some comments on the NASA software defect datasets," *IEEE Trans. Softw. Eng.*, vol. 39, no. 9, pp. 1208–1215, 2013.

[11] S. Agarwal and D. Tomar, "A Feature Selection Based Model for Software Defect Prediction," *Int. J. Adv. Sci. Technol.*, vol. 65, pp. 39–58, 2014.

[12] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11141 LNCS, pp. 270–279, 2018.

[13] E. Kocaguneli, T. Menzies, and E. Mendes, "Transfer learning in effort estimation," *Empir. Softw. Eng.*, vol. 20, no. 3, pp. 813–843, 2015.

[14] Y. Kamei and E. Shihab, "Defect Prediction: Accomplishments and Future Challenges," no. 1, pp. 33–45, 2016.

[15] S. García, J. Luengo, and F. Herrera, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining," *Knowledge-Based Syst.*, vol. 98, pp. 1–29, 2016.

[16] T. Iliou, C.-N. Anagnostopoulos, M. Nerantzaki, and G. Anastassopoulos, "A Novel Machine Learning Data Preprocessing Method for Enhancing Classification Algorithms Performance," pp. 1–5, 2015.

[17] A. Sonali and D. Siddhant, "Prediction of Software Defects using Twin Support Vector Machine", 2nd International conference on Information Systems & computer Networks (ISCON-2014), In press.