SUPPORT VECTOR BASED APPROXIMATION FOR DETECTION OF DEFECT FORECASTING SEVERITY

S.VENKATESH

M.Tech student, CSE Department, JNTUACEA, Anantapur, India
E-mail: svenkatesh343@gmail.com

Abstract— There is non-linear relationship between defects and the software metrics, where complex mapping is being resulted. Therefore, focusing towards defect density, it should be like business requirement of effective and practical approach, which finds the defect density especially in pre-software releases. Soft computing which is a type of evolutionary technique provides a better platform to solve this non-linear and complexity problems. Aim of this proposed technique is to evaluate and also validate a machine learning (ML) approach in prediction of detection of defects. Important constraint of a benchmarking machine learning strategy is to define objective function based on productive universal approximation. Polynet which is based on polynomial machine learning model is used, where it is not similar to the traditional models of machine learning that are based on complexes kernel. It is also framing with the simplified kernel and its objective function is with significant universal approximation. This model is specific to the Industrial Data. Motivation which is given by polynet, redefines kernel strategy and its objective function with universal approximation, where it defines a machine learning model for finding the defect density in pre-software releases. Here support vector machine (SVM) a learning model can also be devised, which uses its objective function with universal approximation for detection of defect severity.

Keywords— Universal approximation, polynomial network, support vector machine.

I. INTRODUCTION

In computer programming, usually while writing source-code of a program there will be possibility of errors and sometimes faults and failures. Presently industrial electronics and other, machine learning (ML) techniques being applied broadly where the domains express high non-linearity and also with dimensionality. Also these machine learning techniques are increased widely in industrial fields by scientists and engineers to get efficiency, rapidness, and accuracy also it computes non-linear problems with these highly known methods. Popular ML algorithms among them like Radial basis function (RBF), support vector regression (SVR), and also fuzzy systems (FSs) will be precisely represented throughout Industrial electronics. It includes like load prediction, power system conditioning, software defect prediction, fault detection and also like manufacturing automation by using RBF variants. This study now presents polynomial-based learning method, which is well prominent than other methods and now Support vector machine (SVM) is devised where it can also be computed for greater efficiency.

In Machine learning, some ideas have been introduced in the previous years in the form of functional link networks (FLNs) which states a single-layer network architecture to the unlimited of function-generating nodes performing summation with output of a single node. Functional-layer network which is having multidimensional inputs X and unlimited number of non-linear functions generating F which will be theoretically approximated to a non-linear function Z. All these multiple nonlinear outputs of function Z would be approximated. This functioning is said to be Universal Approximation.

The objective of the present work is optimizing of non-linear functions which is a part of universally approximation in functional layer network (FLN). Herein, support vector machine (SVM) one of the techniques in ML is used for implementing the single-layer polynomial network, where mostly generated to approximate the multidimensional and non-linear operators. Finally comparing support vector machine performance with rest of the ML techniques for better results.

II. RELATED WORK

In this section area, the functioning of all polynomial networks where it gets simplified to single layer network and used to identify the UA where finally applying on the detection of defect forecasting by devising Support vector machine (SVM)

A. First approach

ML methods or any technique which we propose uses to single type or a multiple related functions as part of computational measures. The replication of these computational measures tends to many advantages like software and hardware implementations. These function approximations and the computational units will hugely determine the ability of representing nonlinearities in defect predictions.

Polynomial machine learning model is getting complicated because of its multi-layer neural networks and also with its recursive nature of structure. It results to lead mainly memory consumption, network complexity, time complexity etc. By observing these deficits, polynet recommending itself where the process should not be monomial and be represented in single layer model with a single matrix layer.

B. Second approach

In this second approach which describes that polynet is recommending the process of network where monomial terms is used, which gets simplified from multiple layer network to single layer or single matrix layer. According to single matrix layer it takes actual constants that is represented from first element to the last in a single row

III. PROPOSED WORK

This approach speaks how universal approximation is computed. The UA value which is defined below is considered. Universal approximation which is being identified by the polynet technique is same applied on the detection of defect forecasting severity by devising support vector machine.

Defining Universal approximation ranging from metric Values:

A. Measuring Metric values

For each metric amount $\{f_i a_j \exists f_i a_j \in fs_{tlr}\}$

Begin

For each metric amount $\{f_k a_l \exists f_k a_l \in fs_{tlr} \wedge j \neq l\}$

Begin

Prepare metric amount pair

$[\{f_i a_j, f_k f_l\} \exists \{f_i a_j, f_k a_l\} \subseteq fs_{tlr}]$ the support of this pair in record set tlr as Records total count, those contain both metric amount values divided by total records

Putting the support as a key amount pair into map fpm_{tlr} as

$$\{f_i a_j, f_k a_l\} \rightarrow s(f_i a_j, f_k a_l)$$

End

End

The same procedure is applied for another metric amount set fs_{flr}.

B. Metric value weight measuring

Measuring further every amount of metric value weight in the form of the respective record set is our next step.

$w_{f_i \leftrightarrow r} \leftarrow 0$ // metric value f_i weight towards record r

For every metric value $\{f_j \exists f_j \in r \wedge i \neq j\}$ Begin

$w_{f_i \leftrightarrow r} \leftarrow w_{f_i \leftrightarrow r} + \{s(f_i, f_j) \exists (f_i, f_j) \in fps$

Combining the weight of all pairs formed by metric value f_i

End

End

$w_{f_i \leftrightarrow tlr} \leftarrow w_{f_i \leftrightarrow tlr} + w_{f_i \leftrightarrow r}$ Combining the weight of metric value f_i towards each record r

End

$w_{f_i \leftrightarrow tlr} \leftarrow \frac{w_{f_i \leftrightarrow tlr}}{|tlr|}$ Averaging weight of metric value f_i finding for record set tlr

End

Maximised procedure according to this section will be adopted for finding metric value weight of the each metric.

C. Measuring Record Weight

Defining the UA of the fault prone and Universal approximation of Inverse of Fault Proneness, measure both the records weight of the respective datasets tlr and flr as follows

Complete process is explored in this area to be noted in tlr that should be used to find record weights of the entire data set flr

D. Metric values measuring and Recording its Influence Weights

The next level scale explains about influence weight $iw_{f->tlr}$ & $iw_{f->flr}$ with each metric value f and also the influence weight $iw_{r\rightarrow tlr}$ & $iw_{r\rightarrow flr}$ with e every record of respective correct record sets tlr and flr measured which is as follows.

For each metric value $\{f \in fs_{tlr}\}$ Begin

$$iw_{f\rightarrow tlr} = \frac{\sum_{p=1}^{|tlr|}\{w_{r\rightarrow tlr} \exists f \in r \wedge r \in tlr\}}{\sum_{p=1}^{|tlr|}\{w_{r\rightarrow tlr} \exists r \in tlr\}}$$

End

For each record $\{r \in tlr\}$ Begin

Averaging the influence weights in resulted metric belongs to the record r as follows

$$iw_{r\rightarrow tlr} = \frac{\sum_{p=1}^{|r|}\{iw_{f_p\rightarrow tlr} \exists f_p \in r \wedge r \in tlr\}}{|r|}$$

End

In the same way influence weight $iw_{f\rightarrow flr}$ of metric amounts $\{f \exists f \in fs_{flr}\}$ of fs_{flr} and also the influence weight $iw_{r\rightarrow flr}$ of every record $\{r \exists r \in flr\}$ of flr is measured.

E. Estimation of the UA Fault Prone Records

Influenced weight records of tlr those measured by the approach further described which is used to define the UA of Fault Proneness as below

$$UA(fp) \leftarrow \frac{\sum_{p=1}^{|tlr|}\{iw_{r_p \to tlr} \exists r_p \in tlr\}}{|tlr|}$$

Fault proneness UA, total average of influence of weights of records in tlr

$$UA(\overline{fp}) \leftarrow \frac{\sum_{p=1}^{|flr|}\{iw_{r_p \to flr} \exists r_p \in flr\}}{|flr|}$$

Inverse fault proneness UA, which is taking average of influence weights of the records in flr

F. Approximating of lower and upper bounds of the UA

For improving prediction accuracy value, define with both the lower, upper bounds of the scales with the help of root mean square deviation method as described below

$$UA_r(fp) = \sqrt{\frac{\sum_{p=1}^{|tlr|}\left(UA(fp) - iw_{r \to tlr}\right)^2}{|tlr|}}$$

$$UA_r(\overline{fp}) = \sqrt{\frac{\sum_{p=1}^{|flr|}\left(UA(\overline{fp}) - iw_{r \to flr}\right)^2}{|flr|}}$$

Lower bounds of the scales are as follow:

$$UA_l(fp) = UA(fp) - UA_r(fp)$$

$$UA_l(\overline{fp}) = UA(\overline{fp}) - UA_r(\overline{fp})$$

In this way the upper bounds of the scales are also defined as:

$$UA_u(fp) = UA(fp) + UA_r(fp)$$

$$UA_u(\overline{fp}) = UA(\overline{fp}) + UA_r(\overline{fp})$$

IV. SUPPORT VECTOR MACHINE

Support vector machines (SVMs) proposed by vapnik and co-workers which is a new type of classifier that is based on statistical form of learning method. Support vector machine are group of supervised learning method that is also being applied to classification. SVMs so far applied to large number of applications like verification, text and face detection, prediction, categorization, speech and speaker verification. This is mainly extension for nonlinear functions. Devising support vector machine (SVM) for detection of defects, it represents at which the specific metric is accurate or to be still improved. Never Fault proneness be zero during the detection of software metrics value. Threshold values is assigned for each metric and during detection if output value is more than the fixed value then it is announced as defect detection is above the high level. Metric values which is less compared to threshold value is announced as defect level is low, when the metric value almost suitable to the fixed value then defect detection will be abnormal.

G. Procedure

Following is the procedure for detection of defect forecasting by inducing support vector machine which is one of the machines learning technique. This method or technique is used as detection of the defect severity with some metric values such as precession and also recall.

H. Programmer

Initially programmer writes the source code. Particular source code which is written by computer programmer is not tested at coding phase due to which may faults or defects. At this stage support vector machine (SVM) is devised for checking the fault proneness with the previous experiences. Basing on results it represent whether the defect level is high or low or abnormal. Results from above are segregated as normal data and fault data, fault data is assigned again for testing where the fault proneness should be decreased to maximum level.

I. Detection of defects with Support vector machine (SVM)

Support vector machine is one of machine learning method. Support vector machine is a type of supervised learning method based on this statistical learning theory which is mainly extension to the nonlinearities that can be applied for segregating or classification. Comparing with other methods for detection of defects, support vector machine is detected with less duration.

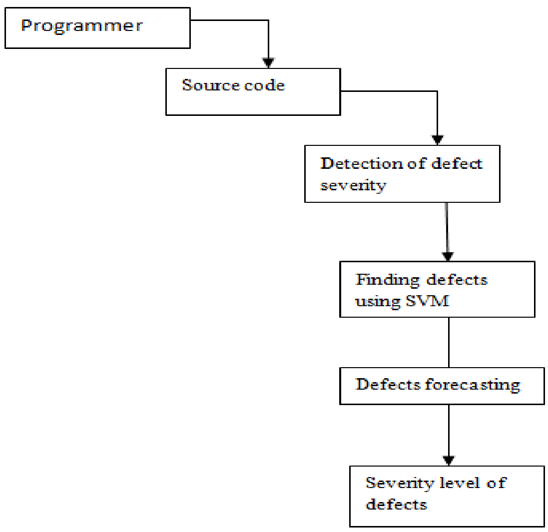

Figure 1: Detection of Defect forecasting by using SVM

V. RESULTS

The fault proneness when compared less than given threshold value, then the defect severity level considered as low, similarly when its value above threshold value then severity is represented as high, if it is equal to threshold value it is abnormal . Taking all java files practically, metrics amounts of fault proneness have been detected as shown below.

The data set used in this proposed model is containing 750 samples which are used to extract the metric values. Further the dataset selected are examined for their fault proneness. Promising of some of the results are shown below in table.

Table 1: Value predicted statistics of the UA

Fault Prone	470
Normal	280
True Positives	460
True Negatives	260
False Positives	20
False Negative	10
Precision	0.958333333
Recall	0.978723404
Accuracy	0.96
Sensitivity	0.978723404
Specificity	0.928571429
F-measure	0.968421053

Table 2: Value predicted statistics of metrics

Fault Prone	470
Normal	280
True Positives	420
True Negatives	200
False Positives	80
False Negative	50
Precision	0.84
Recall	0.893617021
Accuracy	0.826666667
Sensitivity	0.893617021
Specificity	0.714285714
F-measure	0.865979381

The graph is shown below representing accurate results between existing and proposed system**.** Predicted ratios shown in fig 1 with UA are of 460 were predicted and considered to be Fault Prone, and rest 260 are considered as abnormal. All the predicted statistics in table1 are the total value count of false positives is 20 and 10 false negatives, whereas 460 true positives, true negatives got 260. Predicted ratios of CK-Metrics in fig 2 are the count of, 80 false positives and false negatives are 50, whereas true positives resulted 420 and 200 got true negatives.

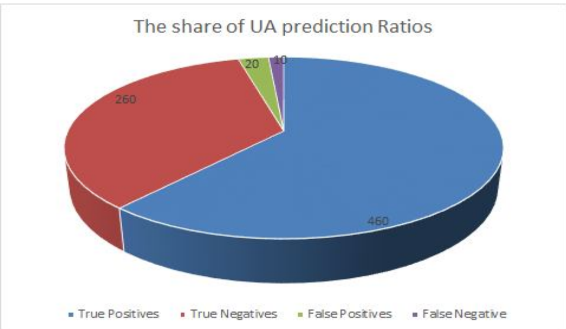

Figure 2: Prediction ratios of Universal approximation

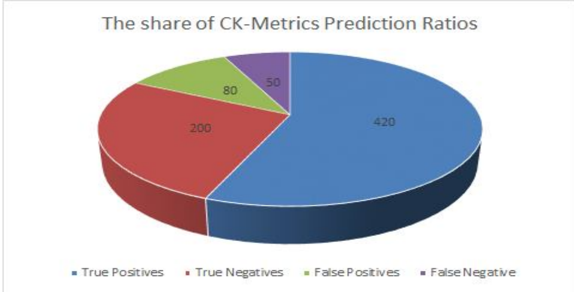

Figure 3: Metrics prediction ratios

Statistical metrics like accuracy, recall, and f-measure are used to assess the prediction accuracy of the UA and CK-Metrics in the above table 2 and 3. The accuracy of UA and CK-Metrics are 96% that means an accuracy got 0.96, accordingly 83% noted as accuracy value noted 0.826 respectively. Now robustness of this model is high level, similarly when sensitivity and recall are observed for Universal approximation which is high when compared to CK-Metrics in fig 3.

In the below graph performance assessment and CK-metrics is shown by comparing under prediction metrics. Accuracy is computed based on only defect severity that has detected in above graph. Based on existing system polynomial network along with universal approximation is used, extending that universal approximation based same as above, towards machine learning which finally applied on defect severity detection.

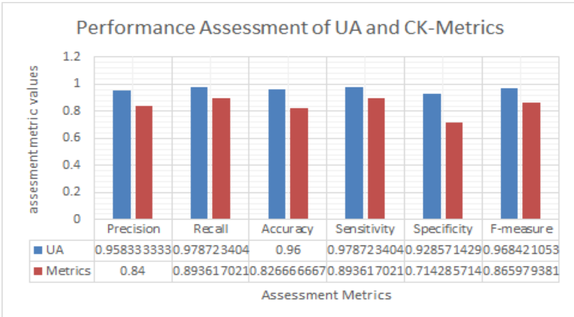

Figure 4: Comparison of Universal Approximation and CK-Metrics under prediction

CONCLUSIONS AND FUTURE WORK

Detection of fault proneness in open source project is also done by devising support vector machine (SVM) which is one type of machine learning technique motivated from the statistical learning theory. It mostly states the art in performing real time applications like image classification, hand written and character recognition, detection of defect severity and also turned into standard tool in machine learning and in data mining. For the detection of defect severity using polynomial network based on universal approximation support vector machine is implemented in simple way. Tested for fault proneness of this open source project which contains of 750 samples. Based on above results support vector machine which is one of the evolutionary technique which makes faster access also by reducing complexity. Therefore, it improved in fault proneness and also the efficiency of a system is achieved overall in future any other machine learning technique or method that can be implemented so that fault proneness could be identified with in less time compared to support vector machine technique.

REFERENCES

[1] Michael S.Pukish, Pawel Rozycki, Bogdan M.Wilamowski "Polynet: A Polynomial-based Learning Machine for Universal Approximation" IEEE Trans. Industrial Informatics, vol.11,no.3, Jun 2015.

[2] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.

[3] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feed forward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[4] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[5] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1– 27:27, 2011.

[6] G. Jekabsons, *GMDH-Type Polynomial Neural Networks*. Riga, Latvia: Riga Technical University, 2011.

[7] S.-K. Oh, W. Pedrycz, and B.-J. Park, "Polynomial neural networks architecture: Analysis and design," *Comput. Elect. Eng.*, vol. 29, no. 6, pp. 703–725, Aug. 2003

[8] V. Vapnik, *The Nature of Statistical Learning Theory*, 1st ed. Hoboken, NJ, USA: Wiley, 1998.

[9] G. Guo, S. Z. Li, and K. L.Chan, Support vector machines for face recognition, *Journal of Image and Vision Computing*, vol. 19, pp. 631-638, (2001).

[10] E. Ardizzone, A. Chella, and R. Pirrone, Pose classification using support vector machines, *In Proceedings of IEEE Int. Joint Conference on Neural Networks*, vol. 6, pp. 317-322, (2000).

★★★