RADIAL BASIS FUNCTION NEURAL NETWORK FOR SOFTWARE ENGINEERING MEASURES - A SURVEY

Umamaheswari E.[1] and D. K. Ghosh[2]
[1]School of Computing Science and Engineering, VIT University, Chennai, Tamil Nadu, India
[2]Department of Mathematics, V.S.B. Engineering College, Karur, Tamil Nadu, India

ABSTRACT

In software quality, the software reliability is the very essential part where it has the capability to manage its individual functions at various conditions. Now-a-days, Software measurements are entirely depends on different techniques like Fuzzy Logic, Neural Network, and Genetic Algorithm etc. This paper reviews SVM (Support Vector Machine) and RBFN to the issues of software measurement in order to increase the correctness as well as the performance. RBF (Radial Basis Function) and SVM has some secure relationship among them where they both are identified in many applications like in face verification, optical character recognition, text categorization and object detection etc. The results examines both the performance analyzes about RBFN and SVM Gaussian Radial Basis Kernel Function. This paper also compares the RBFN and SVM with parameter MRE.

Keywords: software measurement, reliability, neural network.

1. INTRODUCTION

1.1 Introduction of Software measurement

Software Measurement supports the organization in order to develop the measurement program. It doesn't give any assistance for some measurement application like software cost measurement or to observe the difficulties of software. Apart from this, it helps on efficient software measurement program and also to know the key lessons basis on the work [8]. The necessary of the Software Measurement has improved as a well measurement applications. Probability judgment gives accurateness by investigating the effects of two various training methods like performance feedback or environmental feedback. With the help of these feedbacks their individual development in discrimination and calibration is measured and also it is very useful for software development process. Measurement is a method in which symbols or numbers are allocated to the aspect of entities in real world to illustrate the features with described rules. Therefore measurement requires [10].

- Attributes (characteristic of entities)
- Entities (objects of interest).
- Rules (and scales) for allocating values to the attributes.

Software reliability is an essential element in many crucial business applications. Improving reliable software is very hard because of their interdependence between entire software modules like existing software. It is also very hard to identify whether the software which has been delivered is reliable or not. The feedback of the users or clients such as problem reports and system outages compliments or complaints specifies the reliability of various software products [6]. Two types of Software reliability models are there which can support to forecast software reliability by executing the test cases. ”Defect density” models are the initial model in which lines of code, loop, input or output and also the external references are utilized to identify few issues in the software products. ”Software reliability growth models” are the second model. This method is utilized to associate fault detection data with already defined functions like exponential function. This known function is used to predict future behavior if the correlation is fine. Software reliability is explained as the probability for issue free software function for a particular period of time in a particular environment. Software reliability is acknowledged as the most essential factor than the other factors such as maintainability, usability, and functionality [1].

2. REVIEW OF ARTIFICAL NEURAL NETWORK

The survey is based on software measurement presents details to illustrate the measurement importance. This detail contains important activity and procedure for software project measurement and that are named as complexity, quality, cost, and estimation of cost and effort. Software cost is near to software productivity and software quality. In that low cost indicates less project productivity and poor product quality. Here different kinds of techniques are listed below and these techniques are previously used for identifying the reliability of software. The techniques followed below are applied to identify the reliability of software.

2.1 GRNN

The non-linear regression theory is used to support GRNN (General Regression Neural Network) model. Here Y (y1, y2, y3…yn) is the inputs and y is output. E(z|Y) represents expected output value for input Y. f(Y, z) represents density function probability.

$$E(z|Y) = \frac{\int_{-\infty}^{\infty} f(Y, z)\, dz}{\int_{-\infty}^{\infty} z f(Y, z)\, dz} \qquad 1$$

The output of f(Y, z) from training data with size m determined by using a Parzen estimator is follows:

$$z(Y)=E[z|X] = \frac{\sum_{i=1}^{m} W_i \phi_i}{\sum_{i=1}^{m} \phi_i} \qquad 2$$

The ϕi indicates the radial functions, which provide to Radial Basis Function networks. These ϕi functions are called as a kernel functions k(.,.) in literature of kernel methods, here the ϕ links the data with the feature space where as the nonlinear pattern appears as a linear. The ϕi functions can contain various forms with Gaussian, inverse multicuadratic and multicuadratic. Thereforeϕ, i(n) is explained as below:

$$\phi i(n)=\exp\left(\frac{-\| Y(n)-ci\|^2}{2\sigma_i^2}\right) \qquad 3$$

Here $\|Y(n) - ci\|$ represents Euclidean distance among the center ci and input X(n). The spread, standard deviation of Gaussian function is iσ(size of Parzen window). Implementing Gaussian functions as in Equation (2), and explained as [22]:

$$z(Y) = \frac{\sum_{i=1}^{m} W_i \exp\left(\frac{-Di2}{2\sigma_i^2}\right)}{\sum_{i=1}^{m} \exp\left(\frac{-Di2}{2\sigma_i^2}\right)} = \frac{S_w}{S_s} \qquad (4)$$

Di-Euclidean distance and this is similar to Eq.3, and Eq 4 is same as Eqs.5.134 as well as 5.138, this is match to regression estimator of Nadaraya-watson and the normalized Radial Basis Function network [4].

2.2 DENFIS

2.2.1 Basic principles

The DENFIS (A Dynamic Evolving Neural-Fuzzy Inference System) utilizes Takagi-Sugeno kind of engine. This engine is also known as fuzzy inference engine. In DENFIS, engine contains m fuzzy rules specified as follows:

If yi is Rmi, then z is fm(y1,y2,…,yq)
Here yj is Rij, j=1,2,…,q;
i=1,2,3,…,m; are $m \times q$ fuzzy schemes as m precursors for m fuzzy conditions; yj,j=1,2,…q, are precursor variables explained over discourse universes Yj,j=1,2,3,…q, aw well as Rij, i=1,2,3,4,…,m; j=1,2,3,…,q, are fuzzy sets explained by their functions of fuzzy membership.
μ_{Aij}: Yj → [0, 1],
j=1, 2,…,q.
i=1, 2,..,m; in the following parts, y is a subsequent variable, crisp linear functions fi i=1,2,…,m, are applied. In the DENFIS model, all functions is the kind of triangular type illustrated by three parameters as shown in the following equation:

$\mu(y) = mf(y, a, b, c)$ = max(min((y-a)/(b-a),(c-y)/c-b)),0). 3
b- center of cluster on x dimension,
a=b-d*D_{thr},
c=b+ $d \times D_{thr}$
d=1.2~ 2.
D_{thr} - threshold value.
Dthr is a parameter of cluster [9].

2.3 Pi-Sigma network (PSN)

This is a forward highest order of Neural Network, and contains a unique hidden layer. Output layer of PSN contains the product units whereas the hidden layer contains the summing units. Weight which links the input layer and hidden layer are received in training method and the weight which links the neurons of hidden layer with the output layer is permanent to each other. It contains a linear activation process at the hidden layer as well as non linear transfer process at the output layer. In this way the PSN evaluates the product and sum of inputs with equivalent weights and sends it by non-linear process. The Network topology with a single layer of trainable weights radically decreases the time of training. Moreover, by enlarging the input space into highest dimensional space, the product element of Pi-Sigma Network provides the highest abilities. Therefore, PSN simply divides the non-linear separable classes towards linear separable classes. From this, it is very clear that PSN gives Non-linear decision limitations with well classification abilities rather than the linear neuron. [2]

2.4 Multivariates adaptive regression splines (MARS)

MARS gives a stretchy statistical modeling approach which uses backward search and forward search techniques to spot out the integration of basis functions which fits well with data as well as variable selection. MARS is utilized successfully to calculate the unidentified functions in Markov decision method, SDP (Stochastic Dynamic Programming) and Stochastic Programming in optimization. It is also very supportive in many real time optimization issues in which the objective functions are calculated from the data like in simulation optimization [5].

2.5 ANN overview

Ward Network is the initial architecture of neural network. Normally, neural network is the highest communicating network of difficult and huge number of processing components which are called as neurons. Neural network architecture organizes beyond the process on input and output layer as well as it contains one or more intermediately layer known as hidden layer. This input layer neurons is related to hidden layer neurons and it is correlated with the output layer known as hidden output layer weights. An artificial neural network (ANN) is an essential processing system which is similar to the character of biological neural networks. This ANN proceed huge number of highly interconnected processing element known as nodes or neurons or units, in which it usually operates in parallel and it is configured in regular architecture as shown in Figure-1. Every neuron is linked with each other with the help of connection link. The entire connection link is linked along with weights where it includes the details about the input signal. In order to solve a specific problem this details has been used by the neurons net [1].

2.5.1 Architecture of neural network

Based on the study of biological neural network, the artificial neural network is well designed. The brain of the human being is responsible for the entire reactions. These types of reactions are managed by the brain of human and carried out to various parts of human body with the help of rapid processing node called as Neurons. A similar criterion is carried out in the artificial neurons case where it has some rapid processing node which is responsible for managing the computations and they are also called as neurons [7]. The Neural Network is a group of various neurons that are ordered in an efficient manner to manage the computation. The Figure of Neurons basic structure is mentioned below:

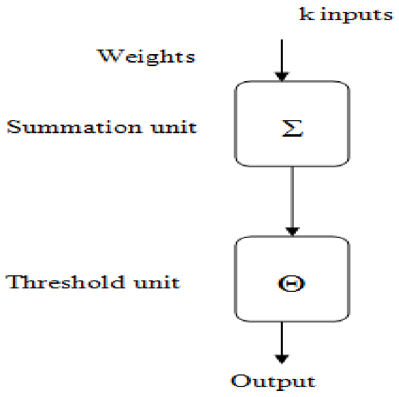

Figure-1. Neuron structure.

The model has been intended for a set of input values and its respective output .let assume that,, I" is the group of input values like I={i1,i2……….ik} and also we connect variable parameter in which it controls the behavior of neurons .Let us assume the weights connected with variable parameter i.e. input like W={w1,w2……….wk}. The entire inputs with weight is passed via the summation unit and this unit finds the $\mathrm{Net}_{\mathrm{input}}$

$$=\sum_{i=1}^{k} I_i w_i$$

The neuron which is arranged by various layers to present a particular structure called as neural network architecture. We require a multilayer of neurons to design a reliability model. These neurons are been arranged in three various layers called as input layer, hidden layer and output layer. In Figure-2, the multi layer networks is shown as follows:

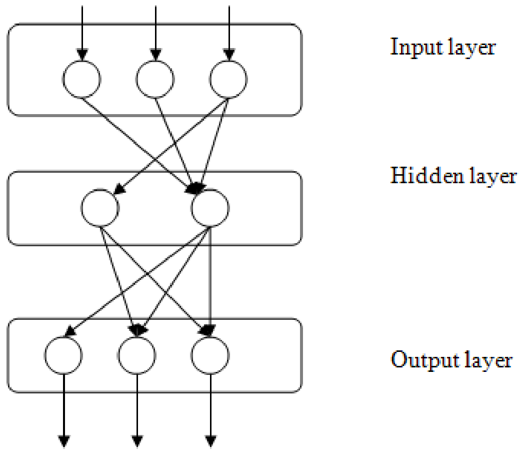

Figure-2. Layered neural networks.

2.5.2 Functioning of artificial neural network: software reliability

Reliability model plays a vital role in software engineering and it is designed for the purpose of failure detection where it has been developed. On the basis of calendar time and execution time, the Software reliability models are designed. Where calendar is nothing but time it takes to complete the task i.e. time interval form the start to end of the program execution on a system. Where as, execution time of a program is denoted as time that a processor takes to execute each instruction of that program.

Artificial neural network or neural network is a group of fast computing and processing artificial neurons known as nodes. These neurons are processing units that are basic to the functioning of the neural network. And these are connected in a definite manner as like layer structure is called as neural network architecture. It is mostly widely used for the purpose of optimizing issues. It can have multiple numbers of layers of processing units in form of feed forward way. Neural network is used as predictor that computes the formal model parameters and discover the process itself. Also have to note that the Back-error propagation is commonly used neural network and it is being used effectively in application studies in wide range of region.

Every neuron has direct connections to the neurons of succeeding layer. Neural networks units have a sigmoid function as an active function in most of the applications. Now a day's most of the research works are being implemented with the interest on Artificial Neural Network software reliability models. Conventionally, both the kinds of the models take only the fault detection process and data for analysis. On the other hand the data that arises from fault correction process and fault detection process are available; ANN and NHPP models are further developed as combined ANN and paired NHPP respectively. Thus it provides more accurate prediction in the process of FDP and FCP [7]. Normally, Analytical approaches are more restrictive when compared with the Data-driven approaches.

3. COMPARTIVE ANALYSIS OF RBFN AND SVM

3.1 Radial basis function network

Universally it is proved that the RBF networks have properties of fast convergence, robustness to outliers and easy solutions to regularization. The Structure of radial basis Function network forms with the three layers, those are input layer, hidden layer and output layer. Here first input layer is constructed with the sources nodes that are linked to the network. Next, hidden layer is formed with the Gaussian function such as radial basis activation function that produces the non linear transformation from input layer to the hidden layer. Final output layer have a linear combination of radial basis function that represent the hidden layer operation in a network with respect to the activation pattern useful on it. Many attractive properties and features of RBF are potential and flexible for the software measurement applications. It is mainly applicable for the issues of pattern classification that is usually solved by modified it into high dimensional space in a nonlinear way. RBF steps are as follow [3].

a) Assign the weight to small random values
b) Until the criterion is false continue the step from 2 to 8
c) For each input of the input layer do the step from 4 to 8
d) Form the each input layer the input signal is send to the next layer i.e. hidden layer
e) Find the Gaussian Function
f) Choose the middle values of radial basis function
g) Compute the output of hidden layer

$$V_i(\chi_i) = \exp[-\sum(\chi_{ij}-\chi^{\wedge}_{ij})^2]/\sigma_i^2 \rightarrow \qquad (1)$$

Here,
$\chi^{\wedge}_{ij}$ = centre of the RBF unit for input variables
σ_i= RBF unit width

h) Compute the neural network output layer output as follow

$$\gamma_n = \sum WI\, V_{i\,(}\chi_i) + W_o \quad \rightarrow(2)$$

Here,
K indicates the number of hidden layer nodes
γ_n indicates the output value for the nth incoming pattern values of mth node in output layer.
Wi indicates weight among mth output node and ith RBF unit.
W_o indicates biasing term at nth output node

i) Final steps are computation of test and error of stopping criterion. This criterion may be number of period otherwise to a definite extent weight modification [1].

3.2 Support vector machines

SVM are the latest method to build the learning machines which can reduce the generalization error. Normally, SVM are very simple with spontaneous ideas. SVM are built by placing a set of hyper planes which divides two or else more data classes. The SVM establishes a limitation among the input classes with the creation of this hyper plane. Support Vectors refers to the components of input data which classifies this limitation. SVM can identify the data which are isolated by the Non-linear limitations. By the usage of kernel function, the problems are mapped to the highest dimensional space where the hyper planes are well enough to classify the limitations.

Consider $[v_{i,}y_i]$ input vector for the labeled training examples, a class value X_i ε{-1,1},
i=1…., where following SVM algorithm defines hyper plane result function for the multi class support classifier

1. Initialize the data that required to be trained under faulty and normal state
2. Construct the formation of SVM and describe the optimal hyper planes with respect to the support vector

$$y_o=\mathrm{sign}(\sum_{i=1}{}^{t}\alpha_i\,(v.v_i)+b)\rightarrow \qquad (3)$$

where, Here y_o is the resultant value,
y_{in} indicates the class value of the training model of $v_{i.}$

The vector v= (v_1, v2…..v_t) represents the input and vectors $v_{i=}$1….t are the support vectors.
α_i and b indicates the parameter that establish the hyper plane.

3. A high-dimensional version of the non-linearly distinguishable case for the equation 3 as follow:

$$y_o=\text{sign}(\textstyle\sum_{i=1}^{t} y_i\, \alpha_{i.}.k\,(v.v_i)+b)\rightarrow \qquad (4)$$

The function $k(v,v_i)$ described as the kernel formula and kernel function for the case of Gaussian function with the kernel function as follow

$$k(v,v_i) =(-\gamma|\, v,v_i\,|^2) \rightarrow \qquad (5)$$

Here, γ represents kernel radial basis function

4. Establish the restraint C by using the cross validation
5. Build the classification function afterwards calculate the SVM model
6. Eliminate the low rank features and end while the features are end [1]

4. RESULTS AND DISCUSSIONS

The PRED (Percentage of the Prediction) and the MMRE (Mean Magnitude of Relative Error) criterion are observed clearly with entire dataset in Table-1. It represents that the RBFN model performs well than the SVM model. Moreover, the SVM performance and the RBFN performance are compared with each other for the development of time prediction issues in this paper.

Table-1. Performance criterion comparison among RBFN and SVM.

Performance Criterion	Prediction Model	
	RBFN	SVM
MMRE	0.108056	0.13508
PRED	0.951219	0.75609

Results represents that the errors are reduced when compared to the original values. It also highlights the significance of Software development time measurement. Figure-3 represents the exact measurement behavior of optimal software development time with applicable datasets which is utilized in this work. The red color fluctuated line shows that the expected development time MRE (Mean Relative Error) of SVM as well as it is compared with the blue color fluctuated line of RBFN in Figure-3. It also represents that both the SVM model and the RBFN model is fluctuated in the region of original development time. The MRE is taken on different observation point in Figure-3.

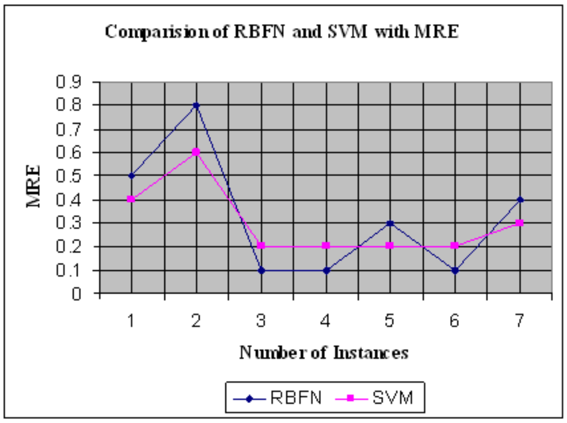

Figure-3. Comparision of RBFN and SVM with MRE.

5. CONCLUSIONS

This paper discussed about the Software Reliability with the help of artificial Neural Networks. It also initiates about the neural model with their architecture. A neural model is to be designed for evaluating the software reliability in future. The Neural Network is an efficient approach which is utilized to measure the output, and it is suitable for managing the datasets which is with smooth tendency rather than for managing the datasets which is with large fluctuations. Normally, the training results are well better than the prediction results. Finally the parameters like MRE, PRED, MMRE are analysed for the prediction model RBFN and SVM.

REFERENCES

[1] Amit Sinhal, Bhavendra Kumar Sinha1 and Bhupendra Verma. A Software Measurement Using Artificial Neural Network and Support Vector Machine.

[2] Ashok Kumar Bhoi, Sibarama Panigrahi and Yasobanta Karali. 2013. A Modified Differential Evolution Algorithm trained Pi-Sigma Neural Network for Pattern Classification.

[3] Boetticher. G. Srinivas. K. 2010. A Neural Net-Based Approach to Software Metrics.

[4] Cuevas-Tello J.C., González-Grimaldo1 R.A. and *et al*. 2012. Parallel Approach for Time Series Analysis with General Regression Neural Networks.

[5] Dachuan T. Shih. 2010. A Convex Version of Multivariate Adaptive Regression Splines.

[6] Garg R. K, Kapil Sharma. 2013. Ranking of software engineering metrics by fuzzy-based matrix methodology.

[7] Gaurav Aggarwal and Gupta V.K. 2013. Neural Network Approach to Measure Reliability of Software Modules: A Review.

[8] Jitendra Singh, Maurya L.S. 2012. Economic Aspects of Reliability Assessment and Prediction of Open Source Softwares - A Review.

[9] Nikola Kasabov and Qun Song. 2009. Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS): On-line learning and Application for Time-Series Prediction.

[10] Ping Cao1, Zhao Dong. 2013. Robust Dynamic Selection of Tested Modules in Software Testing for Maximizing Delivered Reliability.