International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

Finding Successful and Failure of Software using Intelligence Techniques

Janki Sharan Pahareeya, Sanjay Patsariya, Anand Jha, Aradhana Saxena

Department of Information Technology, Rustamji Institute of Technology, Tekanpur, Madhya Pradesh, India

ABSTRACT

This paper presents computational intelligence techniques for software reuse prediction. In this paper, we did comparative study of five computational intelligence techniques that are J-48, Naive-Bayes Classification Algorithm, MLP, random forest and SVM on software reuse data set. We also performed CART based feature selection for reducing the attributes of the data. Ten-fold cross validation is performed throughout the study. The results obtained from our experiments indicate that after feature selection all five techniques were performed well.

Keywords : Vector machine (SVM), Classification and Regression Tree (CART), Multilayer (MLP), J-48, Random forest, software reuse.

I. INTRODUCTION

Data mining is one of the evolution techniques in information technology. It can be named as "knowledge mining from data". Data mining involves many different algorithms to accomplished different tasks. All of these algorithms attempt to fit model to the data and examine the data and determine a model that is closest to the characteristics of the data being examined [1]. The model that is created can be either predictive or descriptive in nature. A predictive model makes a prediction about values of data using known results found from different data. Predictive model data mining tasks include classification, regression and time series analysis etc. Classification maps data into predefined groups or classes [2]. Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown[3,4]. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known). "How is the derived model presented?" The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formula, or neural networks. In this project, selected classification algorithms were considered from Different category of classification algorithms. These algorithms are J48, MLP (Multilayer Perception), RF (Random Forest), Naive-Bayes Classification Algorithm and Support vector machine (SVM) have been considered for comparing their performance based on the Reuse/Predicting successful reuse data.

The rest of this paper is organized as follows. Section 2 related work done in the field of software fault prediction. Section 3 overviews the data description and data preparation, Section 4 overviews of the techniques applied in this paper, section 5 presents the results and discussions. Finally, Section 6 concludes the paper.

II. LITERATURE REVIEW

Faults in software systems continue to be a major problem. They are present in a computer program as errors, flaws, defects, failures, or faults. Over the past years, many fault prediction models have been developed including statistical and machine learning (ML) techniques, of which the machine learning technique is the most popular.

In recent years, a number of alternative modelling techniques have been proposed for the classification of data. They include artificial neural networks, analogy-based reasoning, and fuzzy system [5, 6 and 7] and ensemble techniques. Aggarwal et al. [8] reported an expert committee model, which is a combination of robust regression technique and neural network. Later Vinay kumar et al.[9] reported linear and non linear ensembles consists of various statistical and intelligent techniques viz. Multi Layer Regression, Back Propagation Neural Network (BPNN), RBF, DENFIS, Threshold Accepting based Neural Network (TANN) and SVM. In analogy-based cost estimation, similarity measures between a pair of projects play a critical role [10]. Unfortunately, the accuracy of these models is not satisfactory so, there is always a scope for new prediction technique.

III. DATA DESCRIPTION AND PREPARATION

In our research, we got Reuse/Predicting successful reuse data from PROMISE Software Engineering Repository (http://promise.site.uottawa.ca/SERepository/datasets-page.html). Here you will find a collection of publicly available data sets and tools to serve researchers in building predictive software models (PSMs) and software engineering community at large. Entire Reuse/Predicting successful reuse data set contains information about 24 projects and the data set consist of 29 attributes. In this paper, we are predicting the Success or Failure of the software. We are applying intelligence technique on the data set However, before applying these intelligence techniques to the data set, there are a number of issues to be taken into consideration during cleaning and data preparation. The first cleaning step was to remove the projects having null values. So finally, we got 22 projects in the data set.

IV. OVERVIEW OF TECHNIQUES APPLIED

Naive-Bayes Classification Algorithm:

The Naive-Bayes Classification Algorithm represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems.

We used Weka tool for J-48, Naive-Bayes Classification Algorithm:, MLP ,randem forest and SVM, implementation available at http://www.cs.waikato.ac.nz/~ml/weka/downloading.html

C4.5:

It is an algorithm used to generate a decision tree developed by Ross Quinlan [11] C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, ...$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, ..., x_{p,i})$, where x_j represent attributes or features of the sample, as well as the class in which s_i falls.

This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provides any information gain. In this case, C4.5 creates a decision node

higher up the tree using the expected value of the class.

- Instance of previously unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

The main reason for choosing decision tree learners because it is giving rules that human can easily interpret.

Artificial neural networks are massively parallel interconnections of simple neural that function as a collective system. Neural nets are designed in an attempt to mimic the human brain in order to emulate human performance and thereby function intelligently [12].

A multi-layer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation or training the network [13].MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable.[14]

The multilayer perceptron consists of three or more layers (an input and an output layer with one or more hidden layers) of non linearly- activating nodes. Each node in one layer connects with a certain weight w_{ij} to every node in the following layer. Some people do not include the input layer when counting the number of layers and there is disagreement about whether w_{ij} should be interpreted as the weight from i to j or the other way around.

An advantage of neural nets lies in the high computation rate provided by their massive parallelism, so that real time processing of huge data sets becomes feasible with proper hardware. Information is encoded among the various connection weights in a distributed manner [15, 16].

The SVM is a powerful learning algorithm based on recent advances in statistical learning theory proposed by Vapnik [17] SVM is a learning system that uses a hypothesis space of linear functions in a high dimensional space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. SVR uses a linear model to implement non-linear class boundaries by mapping input vectors non-linearly into a high dimensional feature space using kernels. The training examples that are closest to the maximum margin hyper plane are called support vectors. All other training examples are irrelevant for defining the binary class boundaries. The support vectors are then used to construct an optimal linear separating hyper plane (in case of pattern recognition) or a linear regression function (in case of regression) in this feature space. The support vectors are conventionally determined by solving a quadratic programming (QP) problem.

The new SVM learning algorithm is called Sequential Minimal Optimization (or SMO). Instead of previous SVM learning algorithms that use numerical quadratic programming (QP) as an inner loop, SMO uses an analytic QP step. Sequential Minimal Optimization (SMO) is a simple algorithm that can quickly solve the SVM QP problem without any extra matrix storage and without using numerical QP optimization steps at all. SMO decomposes the overall QP problem into QP sub-problems, using Osuna's theorem to ensure convergence. The advantage of SMO that The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets[18].

V. RESULT & DISCUSSION

In this study we used Reuse/Predicting successful reuse data Set. It has 24 projects with 28 independent variables and 1 dependent variable. On the data sets we applied five intelligence technique that is J-48, Naive-Bayes Classification Algorithm:, MLP ,random forest and SVM on the data sets .The experiments have been carried out two way first way data set is taken at an instance and is supplied to one of the intelligence technique. The parameters of the intelligence technique are set to some initial values and the experiment conducted . Ten-fold cross validation performed for the training the intelligence technique. In this technique, the whole data set is divided into ten parts a nd in first iteration the first nine parts are supplied as training and the tenth part is supplied as testing. In the next iteration, one of the ninth part is taken out for testing and the tenth part is included in the nine parts as training. So, this substitution goes on until all the parts of the dataset have been trained and tested. The results are presented in the table -1 We compared the performance of the Classification models on the basis of Accuracy, which defined as follows:

Accuracy = (True positive + True negative) / (True positive + True negative +False positive + False negative)

The results are presented in Table 1.

Algorithm NAME	ACCURACY(%)
J-48	100
Naive-Bayes Classification Algorithm	100
MLP	100
Randem Foest	95.45
SVM(SMO)	100

In the second way of experiment, we did feature selection. In the field of machine learning feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables) for use in model construction. For the feature selection, we used CART algorithm. By applying Cart algorithm, we got six important variables. Now we did experiment with reduced variable data set. The results are presented in the table -2.

The results are presented in Table 2

Algorithm NAME	ACCURACY (%)
J-48	100
Naive-Bayes Classification Algorithm	100
MLP	100
Randem Foest	100
SVM(SMO)	100

VI. CONSULISION

It is observed that from the table -1 that J-48, Naive-Bayes Classification Algorithm:, MLP and SVM have given 100 5 accuracy without feature selection it is observed that all the techniques are giving 100% accuracy. So we can conclude from the experiments that feature selection should we perform before predicting the success and failure of the software for the Reuse/Predicting successful reuse data.

VII. Acknowledgment

The authors would like to thank the PROMISE Software Engineering Repository for providing the Reuse/Predicting successful reuse data set.

VIII. REFERENCES

[1]. Tribhuvan A.P., Tribhuvan P.P. and Gade J.G. (2015), “Applying Naive Bayesian Classifier for Predicting Performance of a Student Using WEKA. Advances in Computational Research”, ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 7, Issue 1, pp.-239-242.

[2]. Dunham M.H. (2006) “Data mining: Introductory and advanced topics”, Pearson Education India.

[3]. Jankisharan Pahariya, V. Ravi and M. Carr, “Software Cost Estimation Using Computational Intelligent Techniques”, International Conference on Computer Information System and Industrial Management Application, Coimbatore, ISBN: 978-1-4244-5053-4 ,pp.849-854Tamil Nadu, India,2009.

[4]. J.S.Pahariya, V. Ravi, M. Carr and M.Vasu,“ Computational Intelligence Hybrids Applied to Software Cost Estimation”, International Journal of Computer Information Systems and Industrial Management Applications, ISSN: 2150-7988 Vol.2 (2010), pp.104-112.

[5]. S. Andreou and E. Papatheocharous, “Software Cost Estimation using Fuzzy Decision Trees”, Proceedings of 23rd IEEE/ACM International Conference on Automated Software Engineering, 2008, pp. 371-374.

[6]. Mittal, K. Prakash and H. Mittal, “Software Cost Estimation Using Fuzzy Logic”, ACM SIGSOFT Software Engineering Notes, 2010, 35(1), pp. 1-7.

[7]. Attarzadch, “Improving the accuracy of software cost estimation model based on a new fuzzy logic model”, World applied sciences journal, 2010, 8(2), pp. 177-184.

[8]. K.K. Aggarwal, Y. Singh, P. Chandra and M. Puri, “An expert committee model to estimate line of code”, ACM SIGSOFT Software Engineering Notes, 2005, pp. 1-4.

[9]. K. Vinay Kumar, V. Ravi and M. Carr, “Software Cost Estimation using Soft Computing Approaches”,Handbook on Machine Learning Applications and Trends: Algorithms, Methods and Techniques, Eds. E.Soria, J.D. Martin, R. Magdalena, M.Martinez, A.J.Serrano, IGI Global, USA, 2009.

[10]. Y.F. Li, M. Xie and T.N. Goh, “A study of project selection and feature weighting for analogy based software cost estimation”, Journal of Systems and Software, 2009, 82(2), pp. 241–252.

[11]. J. R. Quinlan, “C4.5: Programs for Machine Learning”, Morgan Kaufmann Publishers, 1993.

[12]. Sankar K. Pal and Sushmita Mitra, "Multilayer Perceptron, Fuzzy sets and Classification", IEEE Transection on Neural Networks, Vol. 3No.5,SEptember(1992)

[13]. F.Rosenblatt, “Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms”, Spartan Books, Washington DC, 1961.

[14]. G. Cybenko, “Approximation by superpositions of a sigmoidal function”, Mathematics of Control, Signals, and Systems, Vol.2(1989), PP. 303-314.

[15]. D.E. Rumelhart and J.L. McClelland,Eds.," Parallel Distributed Processing", Vol. 1, Cambridge, MA: MIT press,1986.

[16]. S.E. Fahlman and G.E. Hinton," Connectionist architecture for artificial intelligence", IEEE Computer, PP. 100-109, 1987

[17]. V.N. Vapnik, “Statistical Learning Theory”, John Wiley,New York, 1998.

[18]. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines John C. Platt Microsoft Research.