

Chapter 3.1

Application of Machine Learning Techniques to Predict Software Reliability

Ramakanta Mohanty
Berhampur University, India

V. Ravi
Institute for Development and Research in Banking Technology, India

M. R. Patra
Berhampur University, India

ABSTRACT

In this paper, the authors employed machine learning techniques, specifically, back propagation trained neural network (BPNN), group method of data handling (GMDH), counter propagation neural network (CPNN), dynamic evolving neuro-fuzzy inference system (DENFIS), genetic programming (GP), TreeNet, statistical multiple linear regression (MLR), and multivariate adaptive regression splines (MARS), to accurately forecast software reliability. Their effectiveness is demonstrated on three datasets taken from literature, where performance is compared in terms of normalized root mean square error (NRMSE) obtained in the test set. From rigorous experiments

conducted, it was observed that GP outperformed all techniques in all datasets, with GMDH coming a close second.

INTRODUCTION

Software reliability engineering has become a vital skill for both the software manager and software engineer. The software engineering knowledge is also important to managers and engineers of software oriented products and to users of the products. The primary objective of software reliability engineering is to help the engineer, manager, or users to make more precise decisions. The secondary objective is to make everyone more concretely aware of software reliability by focusing attention on it. Better decisions can save

DOI: 10.4018/978-1-60960-818-7.ch3.1

money on a project or during the life cycle of a piece of software in many ways. In general, the total savings can be more than 10 times greater than the cost of applying these ideas (Musa, 1998).

Reliability is probably the most important of the characteristics inherent in the concept “software quality”. Software reliability is defined as the probability that the software will work without failure for a specified period of time (Musa, 1998). Software reliability is an important factor which is related to defects and faults. It differs from hardware reliability in that it reflects the design perfection, rather than manufacturing perfection. The principal factors that affect software reliability are (i) fault introduction (ii) fault removal and (iii) the environment. Fault introduction depends primarily on the characteristics of the product and the development process. The characteristics of development process include software engineering technologies and tools used the level of experience of the personnel, volatility of requirements, and other factors. Failure discovery, in turn, depends on the extent to which the software has been executed and the operational profile. Because some of the foregoing factors are probabilistic in nature and operate over time, software reliability models have generally been formulated in terms of random processes in execution time.

In the past few years much research work has been carried out in software reliability and forecasting but no single model could capture software characteristics.

In this paper, we investigate the performance of some of the well known machine learning techniques in predicting software reliability. The rest of the paper is organized as follows. In Literature review section, a brief review of the works carried out in area of software reliability prediction is presented. In the next section, the various stand-alone machine learning techniques applied in this paper are briefly described. In the next section, the experimental design followed in this paper is presented. It is followed by a section

that discusses the results obtained. Finally, the last section concludes the paper.

LITERATURE SURVEY

Given the importance of software reliability in software engineering, its prediction becomes a very crucial issue. Machine learning and soft computing techniques have been dominating the statistical techniques in last two decades as far as their applications to software engineering are concerned. The most recent state-of-the-art review, by Mohanty et al. (2010), justifies this assertion. Cai et al. (1991) presented a review on software reliability modeling. The review discussed different types of probabilistic software reliability models and their shortcomings. Then, Karunanithi et al. (1991) employed the BPNN to predict the software reliability and found that the NN models were consistent in prediction and their performance is comparable to that of other parametric models. Later, Karunanithi et al. (1992b) predicted software reliability using a new connectionist approach which offers easy construction of complex models and estimation of parameters as well as good adaptability for different dataset. They used dataset collected from different software systems to compare the models. Based on the experiments, they found that the scaled representation of input–output variables provided better accuracy than the binary coded (or grey coded) representation. The experimental result obtained by them showed that the connectionist networks had less end point prediction errors than parametric models. Thereafter, Karunanithi et al. (1992a) presented a solution to the scaling problems in which they used a clipped linear unit in the output layer. The NN could predict positive values in any unbounded range with a clipped linear unit. They used different types of analytical models such as logarithmic model, inverse-polynomial, power model, delay S-shaped and exponential model. Among the models, the Jordan network model exhibits a better accuracy

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/application-machine-learning-techniques-predict/56151

Related Content

Optimal Sizing of Hybrid Wind and Solar Renewable Energy System: A Case Study of Ethiopia

Diriba Kajela Geleta and Mukhdeep Singh Manshahia (2021). *Research Advancements in Smart Technology, Optimization, and Renewable Energy* (pp. 110-148).

www.irma-international.org/chapter/optimal-sizing-of-hybrid-wind-and-solar-renewable-energy-system/260046

Shape Analysis in Archaeology

Juan A. Barceló (2009). *Computational Intelligence in Archaeology* (pp. 209-228).

www.irma-international.org/chapter/shape-analysis-archaeology/6824

Deep Learning With Analytics on Edge

Kavita Srivastava (2021). *Cases on Edge Computing and Analytics* (pp. 111-133).

www.irma-international.org/chapter/deep-learning-with-analytics-on-edge/271708

A Proposal of SDN Based Disaster-Aware Smart Routing for Highly-Available Information Storage Systems and Its Evaluation

Satoru Izumi, Misumi Hata, Hiroyuki Takahira, Mustafa Soylu, Asato Edo, Toru Abe and Takuo Suganuma (2017). *International Journal of Software Science and Computational Intelligence* (pp. 69-83).

www.irma-international.org/article/a-proposal-of-sdn-based-disaster-aware-smart-routing-for-highly-available-information-storage-systems-and-its-evaluation/175656

Modified Backward Chaining Algorithm Using Artificial Intelligence Planning IoT Applications

Pradheep Kumar K. and Srinivasan N. (2019). *Edge Computing and Computational Intelligence Paradigms for the IoT* (pp. 153-169).

www.irma-international.org/chapter/modified-backward-chaining-algorithm-using-artificial-intelligence-planning-iot-applications/232006