International Journal of Information Technology and Management

Vol. IX, Issue No. XIV, November-2015, ISSN 2249-4510

ANALYSIS ON CODE OPTIMIZATION USING DATA MINING APPROACH

AN INTERNATIONALLY INDEXED PEER REVIEWED & REFEREED JOURNAL

www.ignited.in

International Journal of Information Technology and Management
Vol. IX, Issue No. XIV, November-2015, ISSN 2249-4510

Analysis on Code Optimization Using Data Mining Approach

Arjun Singh

B.Tech, IV Year, Northern India, Engineering College, Delhi, India

Abstract – Billions of lines of code are currently running in Legacy systems, mainly running machine critical systems. Large organizations and as well as small organizations extensively rely on IT infrastructure as the backbone. The dependability on legacy Software systems to meet current demanding requirements is a major challenge to any IT profession. One of the top priority of any IT manager is to maintain the existing legacy system and optimize modules where required. Various techniques have been developed to determine the complexity of the modules as well as protocols have developed to assess the severity of a software problem.

In this paper, it is proposed to study data mining algorithms in a multiclass scenario based on the severity of the error in the module.

Keywords - Legacy Software, Normalization, Data Mining, Random Tree, Bayesian Logistic Regression

----------X----------

1. INTRODUCTION

Legacy systems are older software system and typically its original designers and implementers are no longer available to perform the system's maintenance. Often specifications and documentation for a legacy system are outdated / not available, so the only definitive source of information about the system is the code itself.

Organizations can have compelling reasons for keeping a legacy system, such as: The system is able to cope up with the current requirements and the management sees no need to change it. Cost of Retraining and lost time would be high. (Meacham, et.al., 2009)

Frequently legacy systems are expensive to maintain and upgrade and have extreme limitations of function. They do not interface with new technologies well and available pool of support resources is dwindling. They are considered to be potentially problematic by many software engineers for several reasons Legacy systems often run on obsolete hardware, and spare parts for such computers may become increasingly difficult to obtain. (Hunold, et.al 2008)

- The cost of maintaining the system will eventually outweigh the cost of replacing both the hardware and software
- Integration with newer systems may also be difficult because new software may use completely different technologies.

The kind of bridge hardware and software that becomes available for different technologies that are popular at the same time are often not developed for differing technologies in different times, because of the lack of a large demand for it and the lack of associated reward of a large market economies of scale, though some of this "glue" does get developed by vendors and enthusiasts of particular legacy technologies (Kangtae Kim, 2007). Where it is impossible to replace legacy systems through the practice of application retirement, it is still possible to enhance them. Most development often goes into improving the code of a legacy system.

Various computation methods have evolved with increasing number of lines of code, which can give accurate. This data can be utilized by extracting knowledge using Data Mining techniques. (Fayyad, et.al)

Data mining becoming a very important tool already used in business intelligence, marketing intelligence, machine vision, genetic engineering, biotechnology and so on. Lot of research on data mining applicable to specific domains is being carried out by researchers in academic circles and in industries alike. For validation of algorithms and their effectiveness against known standards become extremely important. The need to compare results

against a fixed a standard data set becomes important. Various organizations have assimilated huge repositories of datasets, which can be used as a standard for validating algorithms, and at the same time compare proposed systems with existing systems. In this research software reliability is focused, by proposing to use the NASA dataset and in particular the KC1 data set for classification and reliability prediction. (Guo. L et.al., 2003)

2. REVIEW OF LITERATURE

An IDS using data mining approaches was proposed by (Lee and Stolfo 1996 and 1998). They have suggested that the association rules and frequent episodes algorithms can be used to compute the consistent patterns from audit data. This method provides the basis for feature selection and was used to discover patterns of intrusions. A number of approaches based on system calls were proposed. Forrest et al (1997) illustrated how the principles of human immunology could be incorporated into a computer intrusion detection framework. A database of normal sequences of system calls was built and those sequences which were not found in the database were considered as anomalies. In their work, an approach for a host based anomaly detection called time-delay embedding (tide) is proposed, wherein traces of normal application executions were noted. A sliding look-ahead window of a fixed length was used to record correlations between pairs of system calls. The correlations were stored in a database of normal patterns, which was then used to monitor sequences during the testing phase. (Han J. and M. Kamber,) Anomalies were accumulated over the entire sequence and an alarm was raised if the anomaly count exceeded the threshold. Tide forms correlations between pairs of system calls within a certain preset window size. Evidence is given that short sequences of system calls executed by running programs are a good discriminator between normal and abnormal operating characteristics of several common UNIX programs. It is experimentally proved that the definition is stable during normal behavior of standard UNIX programs. Further, it is able to detect several common intrusions involving send mail and lpr. These provided the basics for later intrusion detection systems.

26Here, the machine is trained with known patterns. When a test pattern is introduced, intrusion is detected if it deviates from the trained one by a larger extent. Lee et al (1997) demonstrated that ‘machine learning’ approach can be used to learn normal and/or abnormal patterns from the data. This approach extended Forrest’s work by making use of a generalization algorithm and proved the effectiveness of converting short system call sequences into a set of rules. Ryan (1998) used neural networks in detecting intrusion detection. He employed Self Organizing Maps, a type of neural network in his approach. The major advantage of using Self Organizing Maps is that it maps the high dimensional data over a low dimensional featured space. Hence, the behavior set that used was a multi-dimensional one. So in this technique it is able to map the behavior set to two groups of low dimensional one. Here one group represents normal events and the other involves intrusion events.

Ghosh et al (1999) proved that anomaly detection using Elman networks gave a better performance compared to profile–based techniques. The system call data had to be mapped into feature space and the choice of feature space was based on the application. As the system-call data is fine grained, overhead increases thereby decreasing the system performance. This throttled research on ‘anomaly detection by modeling file activities’. The direct clustering methods for IDS are discussed by Vesanto and Alhoniemi (2000). The SOM (Su and Chang 2000) is an excellent tool in the exploratory phase of data mining. It projects input space on prototypes of a low-dimensional regular grid that can be effectively used to establish utilized to visualize and explore properties of the data. When the given number of SOM units is large, to facilitate quantitative analysis of certain the map and the data, similar units need to be grouped, i.e., clustered in (Xu and Chow 2010, Havens et al 2010). Liu and Wang (2008) proposed integrated intrusion detection system using multiple neural networks. This includes principal component neural networks and principal component self-organizing map networks. It is able to detect the intrusions/attacks both from the outer internet and an inner LAN. In Irfan et al (2009) work a data clustering algorithm for supplier categorization namely S-Canopy clustering was proposed. It is simply making use of canopy clustering to reduce the number of distance comparisons. This enhances the classification results while used in the anomaly detection.

Anderson et al (1995) proposed Statistical Component of the Next- Generation Intrusion Detection Expert System (NIDES) in detecting unusual program behavior. Wagner and Soto (2002) constructed a Non-Deterministic Push Down Automata (NDPDA) to represent program behavior and evaluated the proposed framework for host-based IDS against mimicry attacks. Sequeira and Zaki (2002) designed and implemented a temporal sequence clustering- based intrusion detection which is user-profile dependent. This was done by collecting and processing UNIX shell command data. Though analyzing a user activity is a natural approach to detect intrusions, experience shows that it is far from accurate. It is because of the fact that user behavior typically lacks strict patterns. Dynamic usage by user gave a greater reflection of the features that define host behavior. In their work, user activity involves using programs. Programs obtain the required services by executing the specific system calls that provided the needed function. The sequence of system calls executed by a program should be regular and predictive, as the code of a given application should not change. Liao and Vemuri (2002) used frequencies of system calls to

model system behavior and used the k-Nearest Neighbor classifier to detect aberrations. This methodology has computational advantages over profile-based approaches.

3. DATA NORMALIZATION

It is proposed to use the Bayesian logistic regression classification algorithm and compare the result with Random tree classification algorithm and CART using weka. (A Machine Learning-Based Reliability, 2007)

Random trees are formed by a stochastic process. Random binary tree are binary trees with a given number of nodes, formed by inserting the nodes in a random order or by selecting all possible trees uniformly at random. Random trees can also be formed using spanning methods. (Hongyu Zhang, et.al., 2007)

Classification and regression trees (CART) is a non-parametric technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively.

Trees are formed by a collection of rules based on values of certain variables in the modeling data set. (Sheng Yu, 2010)

- Rules are selected based on how well splits based on variables' values can differentiate observations based on the dependent variable
- Once a rule is selected and splits a node into two, the same logic is applied to each "child" node. (Riyad Alshammari and A. Nur Zincir)
- Splitting stops when CART detects no further gain

Databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several Giga bytes or more) and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low- quality mining results. There are number of data processing techniques. Data cleaning is one that can be applied to remove noise and collect inconsistencies in the data. Data integration merges data from multiple sources into a coherent data store, such as data warehouse. Data transformation, such as normalization, may be applied. Normalization may improve the accuracy and efficiency of mining algorithms involving distance measurement. Data reduction can reduce the data size by aggregating, eliminating redundant features, or clustering, for instance. These techniques are not mutually exclusive; they may work together. Data cleaning can involve transformations to correct wrong data, such as by transforming all entries for data field to a common format. Data processing techniques, when applied before mining, can substantially improve the overall quality of the pattern mined and/or not the time required the actual mining.

In this paper data is pre-processed using mathematical model i.e. Normal Density Function.

The equation for Normal Density Function (Cumulative = False)

When cumulative is true, the Formulae is can be made, or some pre-set stopping rules are met. Each branch of the tree ends in a terminal node

- The equation for Normal Density Function (Cumulative = False) is given by f(x, μ, σ) = $1/\sqrt{2\Pi}\times\sigma * e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma}}$.

When cumulative is True, the Formulae is $\int_{-\infty}^{x}\frac{1}{\sqrt{2\Pi}\times\sigma}\times e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

Normdist (x, mean, std, cumulative).

The meaning of the above function is as follows.

If cumulative is time Normdist function returns cumulative distribution function, otherwise it returns the probability mass function.

- x is the value of the attribute which we need to find the distribution.
- Mean is an arithmetic mean of the distribution.
- Std is the standard deviation of the distribution.
- Cumulative is a logical value which determines the form of the function.

The attributes used in this work is described briefly below

LOC_BLANK - The number of blank lines in a module.

LOC_CODE_AND_COMMENT - The number of lines, which contain both code & comment in a module.

LOC_COMMENTS - The number of lines of comments in a module.

CYCLOMATIC_COMPLEXITY - The cyclomatic complexity of a module.

DESIGN_COMPLEXITY - The design complexity of a module.

ESSENTIAL_COMPLEXITY - The essential complexity of a module.

LOC_EXECUTABLE - The number of lines of executable code for a module (not blank or comment) HALSTEAD_CONTENT - The halstead length content of a module.

HALSTEAD_DIFFICULTY - The halstead difficulty metric of a module. HALSTEAD_EFFORT - The halstead effort metric of a module.

HALSTEAD_ERROR_EST - The halstead error estimate metric of a module. HALSTEAD_LENGTH - The halstead length metric of a module.

HALSTEAD_LEVEL - The halstead level metric of a module. HALSTEAD_PROG_TIME - The halstead programming time metric of a module. HALSTEAD_VOLUME - The halstead volume metric of a module.

NUM_OPERANDS - The number of operands contained in a module. NUM_OPERATORS - The number of operators contained in a module.

NUM_UNIQUE_OPERANDS - The number of unique operands contained in a module.

4. APPROACHES TO HOST-BASED INTRUSION DETECTION SYSTEM

When an attack takes place, attackers usually replace critical system files with their versions to inflict damage. Tripwire is an open-source host-based tool, which performs periodic checks to determine which files are modified in the file system. To do so, Tripwire takes snapshots of critical files. A snapshot is a unique mathematical signature of the file where even the smallest change results in a different snapshot. If the file is modified, the new snapshot will be different than the old one; therefore critical file modification would be detected. Tripwire is different from the other intrusion detection systems because rather than looking for signs of intrusion, Tripwire looks for file modifications.

CONCLUSION

The Results tabulated in fig. 1shows that data mining is a viable method to predict modules that require optimization. The knowledge mined can be the starting point to the any IT manager to plan his strategy for software maintenance. The results can be validated for different data sets to establish uniformity of our proposed solution.

REFERENCES

A Machine Learning-Based Reliability Assessment Model for Critical Software Systems Challagulla, V.U.B. (2007) Computer Software and Applications Conference.

A Preliminary Performance Comparison of Two Feature Sets for Encrypted Traffic Classification Riyad Alshammari and A. Nur Zincir- Heywood Dalhousie University, Faculty of Computer Science And Uthurusamy, R. (eds.), AAAI Press.

A survey on metric of software complexity , Sheng Yu; Shijie Zhou; (2010) The 2nd IEEE International Conference on Information Management and Engineering (ICIME),

Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth, P., From Data Mining to Knowledge Discovery: An Overview. In Advances in Knowledge Discovery and Data Mining, Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (eds.).

Han J. and M. Kamber, Data Mining: Concepts and Techniques, 2nd edition. Morgan Kaufmann,

Hunold, S.; Korch, M.; Krellner, B.; Rauber, T.; Reichel, T.; Runger, G, (2008). Transformation of Legacy Software into Client/Server Applications through Pattern-Based Rearchitecturing Computer Software and Applications,. COMPSAC '08. 32nd Annual IEEE International Digital Object Identifier:10. 1109/COMPSAC. 2008. 158, Publication Year: 2008, Page(s): pp. 303 – 310.

Kangtae Kim; Hyungrok Kim; Woomok Kim; (2007). Building Software Product Line from the Legacy Systems "Experience in the Digital Audio and Video Domain", Software Product Line Conference, 2007. SPLC 11th International Digital Object Identifier: 10.1109/SPLINE.2007.27 Publication Year: 2007.

Meacham, D.J.; Michael, J.B.; Man-Tak Shing; Voas, J.M, (2009). Standards interoperability: Applying software safety assurance standards to the evolution of legacy software, System of Systems Engineering, 2009. SoSE. IEEE International Conference, Publication Year: 2009 , Page(s): pp. 1 – 8.

Predicting Defective Software Components from Code Complexity Measures Hongyu Zhang; Xiuzhen Zhang; Ming Gu; PRDC (2007). 13th Pacific Rim International Symposium on Dependable Computing.

Predicting fault prone modules by the Dempster-Shafer belief networks Guo, L. Cukic, B. Singh, H. (2003) Lane Dept. of CSEE, West Virginia Univ., Morgantown, WV, USA , 18th

IEEE International Conference on Automated Software Engineering,.