Indian Journal of Science and Technology, Vol 9(4), DOI: 10.17485/ijst/2016/v9i4/52931, January 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Analysis of Process Mining Model for Software Reliability Dataset using HMM

V. Priyadharshini* and A. Malathi

PG and Research Department of CS, Government Arts College, Coimbatore – 18, Tamil Nadu, India; priyaphd13@gmail.com, malathi.arunachalam@yahoo.com

Abstract

Process mining analyzes business processes constructed on event logs. To derive a process model from a log of recorded events, HMM and reliability metrics are used to derive the required petri net. Two software reliability datasets are used to derive petri nets. This dataset describes particular software's failure cycle. Using this failure cycle, its corresponding event log is recorded. HMM and software reliability are used to track recorded event logs.

Keywords: Helix and RALIC Dataset, Process Mining, Process Discovery, Reliability Metrics, Software Reliability

1. Introduction

Software reliability is directly related to system reliability. Software reliability cannot be guaranteed by redundancy, and methods to verify its reliability are not like hardware which has a complete theoretical system. Event-based parameters like Petri nets, model event causality, conflict, and concurrency, thus, providing alternative information to state-based models. It is often captured in a more concise form[2]. The theory of regions offers a bridge between state and event-based representations. This issue contains transforming a state-based into an event-based parameter through preserving the behavior. In particular, the theory of regions was designed for transforming a Transition System (TS) into a Petri net. The theory is primarily defined for uncomplicated transition systems deriving 1-bounded Petri nets, whereas this restriction was ignored.

Process discovery the most stimulating process mining tasks. Based on event log, a process model is built by capturing its behavior of the log[3]. Event logs basically capture the business activities happened at a certain time period[4]. The basic plan is to extract data from event logs recorded by a data system. This aims by providing procedures and tools for locating method, control, data, structure, and social configurations from event logs. The research area that is concerned with knowledge discovery from event logs is called process mining[5]. New techniques are established to perform process mining i.e. mining of process models. It is the traditional analysis of business processes based on the opinion of process expert[6]. The business process mining attempts to modernize a complete process model from data logs that contain real process execution data. Many techniques focus the possibility of combining a number of process mining approaches to mine more stimulating event logs, such as those that contain noise.

Caccia in[7] describes the design and implementation of an execution control module which, through suitable graph-search algorithms, generates orders of task activation/deactivation operations which execute the desired commands preserving the system in admissible configurations. Zouaghi in[8] describes a recursive and nested hybrid monitoring and diagnosis architecture for systems with Recursive Nested Behavior-based Control structure. Such systems consist of behavioral levels, which use several models on different levels of abstractions. the monitors of each subsystem use recursively the output of the monitors of the next lower level in order to get an estimate of the global status of the system at each time and having

* Author for correspondence

the advantage of low dimensionality for each level. Ernst in[9] discusses the generalization of the basic reachability tree construction which is made symmetric with respect to the first and last marking. Sets of transition sequences defined by finite automata are used for calculations to notice orders, and the approximation error is assessed by Presburger expressions. The approximation algorithm is repeated until a necessary criterion for reachability is satisfied. Manuel Silva in[10] describes the fluidization of discrete event dynamic models, an efficient technique for dealing with the classical state explosion problem. It centers on the relationships among distinct and constant PN models, both for untimed, i.e., fully non-deterministic concepts, and timed versions; the use of structure theory of (discrete) PNs, algebraic and graph based ideas and consequences and the bridge to Automatic Control Theory.

2. Dataset Description

The two input datasets we have taken is referred as Helix and Ralic. Ralic dataset holds 262 weighted attributes across 10 requirement sets from 79 stakeholders. Helix dataset contains releases and metrics. The JAR files consist of class files for each release with meta data. A metric history is obtained from extraction of the releases.

3. Proposed Method

The main contribution of our research is considering the frequencies, this is used to distinguish between real and noisy states, because the second have frequently low frequency. For example, only vector differences between frequent states could be taken into account to discriminate real folding opportunities from false cycle unfolding produced by noise[11].

To get more robust to noise while tracking the log event, we use Hidden Markov Model (HMM). This is suitable for frequency tracking of a signal. By using this it is considered that it is used to distinguish between real and noisy states, because the latter have often low frequency. Also it has capability of handling the noise and unreliable data[12]. HMM includes finite number of states with predefined state transition probabilities. The probabilities of particular states are according to existing state only. Each and every state belongs to diverse variable set which can be observed[16]. The variable state cannot be examined straightly in this HMM method because every variable state is associated with specific observation probabilities. This means possibilities of observing a particular variable set. Among several algorithms like Viterbi or the Forward-Backward algorithm we can achieve an optimal state sequence in the HMM method[14]. When information concerning the frequency evaluation method and SNR are incorporated in the observation probabilities, a prior knowledge of likelihood in which the frequency transforms are added in the state of transition probabilities[13]. For example, only vector differences between frequent states could be taken into account to discriminate real folding opportunities from false cycle unfolding caused by noise. We propose a Hidden Markov Model (HMM) which is well-suited for frequency tracking of an event log[10]. The states of the model relate to the actual frequency, while the observations correspond to the estimated frequency of a specific time interval of the event log. A prior knowledge of the likelihood by which the frequency changes is included in the state transition probabilities, while knowledge about the frequency estimation method and the Signal-to-Noise Ratio (SNR) are included in the observation probabilities[15]. Experiment results suggest that our proposed technique is an accurate for evaluating the frequency tracking of an event log through HMM.

3.1 Emission Probability

It does not change over time.

$$p(x_n \mid z_n, \varnothing) = \prod_{k=1}^{K} p(X_n \mid \varnothing_k)^{z_{nk}} \tag{1}$$

3.2 Transition Probability

The probability of going from a given state to the next state is defined as transition probability.

Probability of going from state j to state k:

$$A_{jk} \equiv p(z_{nk} = 1 \mid z_{n-1}, j = 1)$$
$$\sum_k A_{jk} = 1 \tag{2}$$

Probability of state k being initial state

$$\pi_k \equiv p(z_{1k} = 1)$$
$$\sum_k \pi_k = 1 \tag{3}$$

Based on the above three probabilities the joint probability is obtained, which while deriving petri net will be used and the required petri net is achieved.

3.3 Reliability Metrics

Reliability metrics are units of measure for system reliability. It is measured by counting the number of operational failures and relating these to demands made on the system at the time of failure. A long-term measurement program is required to assess the reliability of critical system. It is used to quantitatively express the reliability of the software product[16]. The choice of which metric is to be used depends upon the type of system to which it applies and the requirements of the application domain. Some reliability metrics are discussed below:

Mean time to failure:

It is calculated based on operating system, number of cycles and calendar time. It describes expected time to failure. It is difference of time between two consecutive failures. It is relevant of systems when individual transactions take lots of processing time. It is calculated by,

$$\text{MTTF} = \frac{(\text{Time period * number of items tested})}{\text{Number of items tested within time period}}$$

Mean time to repair:

It is a measure of mean time between the points at which the failure is first discovered until the point at which it returns to operation. It is the time required to fix a failure. It is calculated by,

$$\text{MTTR} = \frac{\text{Total corrective maintenance time}}{\text{Total number of corrective maintenance}}$$

Mean time between failure:

It is predicted time between inherent failures of a system during operation. It is calculated by,

$$\text{MTBF} = \frac{\text{Sum of operational failures}}{\text{Number of observed failures}}$$

To evaluate the quality of a petri net, with HMM based techniques, first a mapping has to be done from petri nets to HMM. Initially create a HMM based on given petri net, then relate the sequences from event log to HMM to evaluate appropriate match.

4. Results and Discussion

In this work HELIX and RALIC dataset are used. The fraction of unconnected transition and time complexity rate for probability finding is calculated based on petri net generation is shown in Figure 1. The proposed method has highest efficiency than the existing system.

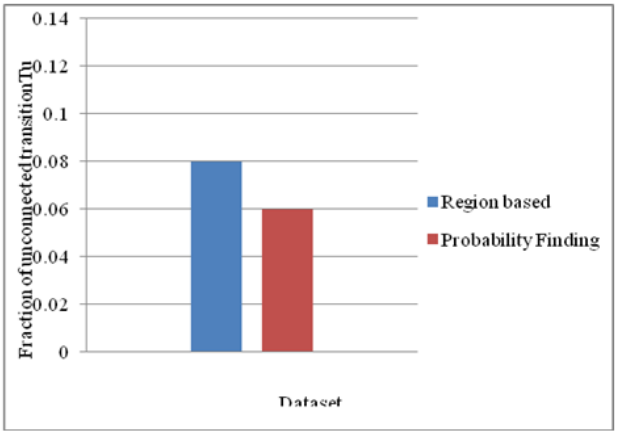

Figure 1. Fraction of unconnected transition T_u.

We calculate T_u by comparing it with existing region based method and proposed probability finding method. It is decreased for the proposed method thus it helps in increasing the performance of proposed method.

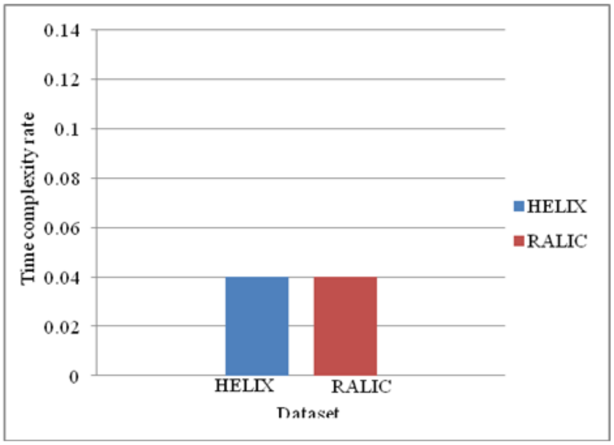

Figure 2. Time complexity rate.

Figure 2 describes time complexity rate, i.e. time taken to execute a particular task within specified time. If the time complexity rate is low, it means the performance of the proposed system is running efficiently.

Table 1. Values for average metrics against number of traces

No. of iterations	HELIX	RALIC
1	76.1900	75.1900
2	78.5700	77.5700
3	80.9500	79.9500
4	85.7100	83.7100
5	90.4800	89.4800
6	95.2300	93.2300

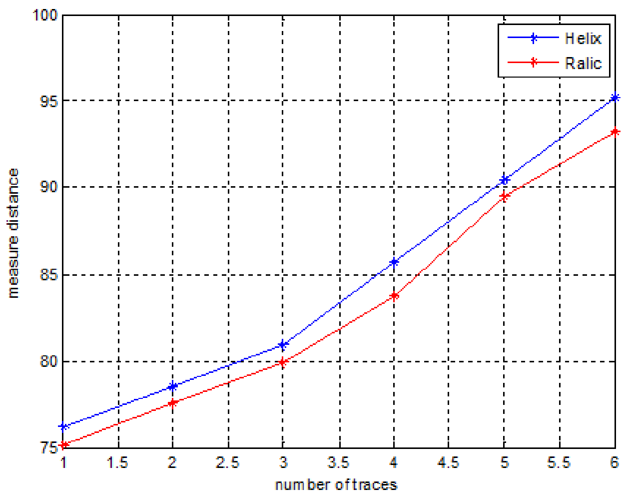

Figure 3. Average metrics against number of traces.

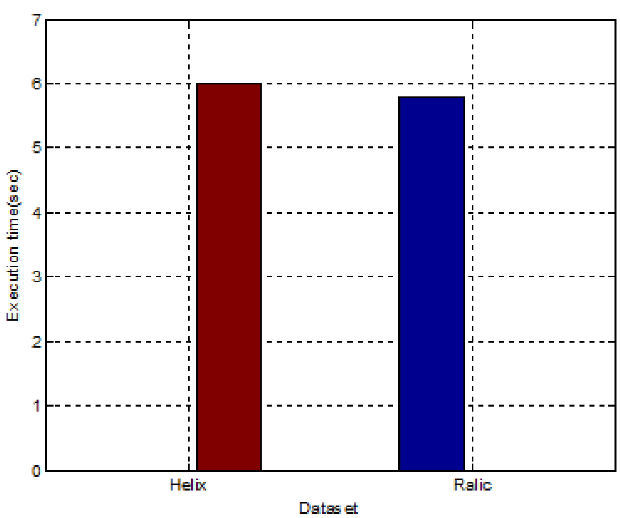

Figure 4. Execution Time in Seconds.

The values for average metrics against number of traces are given in Table 1, and its corresponding graph is shown in Figure 3. The execution time for each dataset used is given in Figure 4. In Figure 5, the petrinets execution time in seconds is shown.

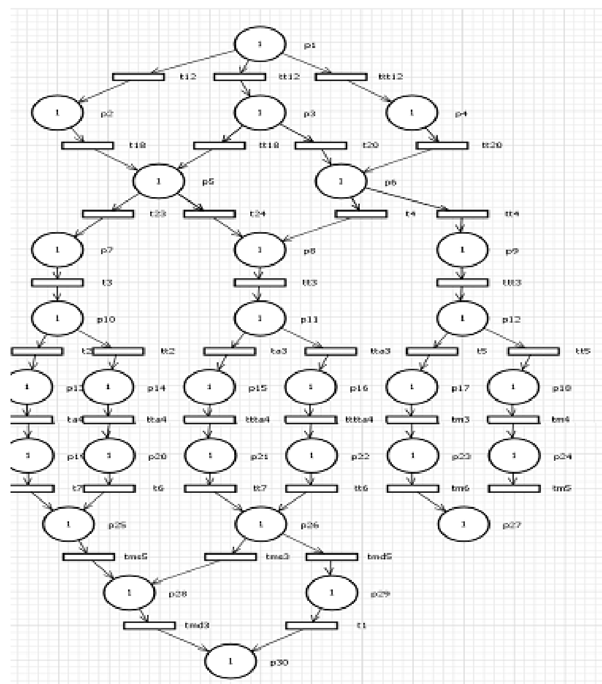

Figure 5. Petri net execution for probability finding.

5. Conclusion and Future Work

The datasets are used with two proposed methods. The existing method improves and allows models based on realistic characters of event logs. Compared to other region based methods, it yields feasible and effective solution to derive petri nets. It is used to find optimal solution at a faster rate. The proposed method increases the chances to find optimal solution as it decreases with time.

6. References

1. Yethiraj NG. Improvement of Software Maintenance and Reliability using data mining techniques. International Journal of Data Mining Techniques and Applications. 2012; 1(2):101–6.
2. van der Aalst WMP, Weijters AJMM, Maruster L. Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering. 2004; 16(9):1128–42.
3. Leemans SJJ, Fahland D, van der Aalst WMP. Discovering block-structured process models from event logs- A constructive approach. Application and Theory of Petri Nets and Concurrency. Lecture Notes in Computer Science. 2013; 7927:311–29.
4. Bae J, Liu L, Caverlee J, Zhang L-J, Bae H. Development of distance measures for process mining, discovery and integration. International Journal of Web Services Research. 2007; 4(4):1–17.

5. van der Aalst W, Adriansyah A, de Medeiros AKA, Arcieri F, Baier T, Blickle T, Bose JC, et al. Process Mining Manifesto. Business Process Management Workshops. Lecture Notes in Business Information Processing. 2012; 99:169–94.
6. Caccia M, Coletta P, Bruzzone G, Veruggio G. Execution control of robotic tasks: A Petri net-based approach. Control Engineering Practice. 2005; 13(8):959–71.
7. Zouaghi L, Wagner A, Badreddin E. Hybrid, recursive, nested monitoring of control systems using Petri nets and particle filters. International Conference on Dependable Systems and Networks Workshops (DSN-W), Chicago. 2010; 73–9.
8. Mayr EW. An Algorithm for the General Petri Net Reachability Problem. SIAM Journal of Computing. 1984; 13(3):441–60.
9. Silva M, Julvez J, Mahulea C, Renato Vazquez C. On fluidization of discrete event models: observation and control of continuous Petri nets. Journal of Discrete Event Dynamic Systems. 2011; 21(4):427–97.
10. Venkataramanan L, Sigworth FJ. Applying Hidden Markov Models to the Analysis of Single Ion Channel Activity. Biophysical Journal. 2002; 82(4):1930–42.
11. Kereliuk C, Depalle P. Improved hidden markov model partial tracking through time-frequency analysis. Proceedings of International Conference on Digital Audio Effects (dafx-08), Espoo, Finland. 2008.
12. Rozinat A, Veloso M, van der Aalst WMP. Evaluating the Quality of Discovered Process Models. Proceedings of Induction of Process Models. IPM workshop at ECML PKDD. 2008; 45–52.
13. Paris S, Jauffret C. Frequency line tracking using hmm-based schemes. IEEE Transactions on Aerospace and Electronic Systems. 2003; 39(2):439–49.
14. Schliep A, Schonhuth A, Steinhoff C. Using hidden Markov models to analyze gene expression time course data. Oxford Journals. 2003; 19(1):255–63.
15. Kaur G, Bahl K. Software Reliability, Metrics, Reliability Improvement using Agile Process. International Journal of Innovative Science, Engineering and Technology. 2014; 1(3):143–7.