

GUSTAVO DA MOTA RAMOS

Seleção de técnicas de geração automática
de dados de teste por meio de aprendizado
de máquina

São Paulo

2016

GUSTAVO DA MOTA RAMOS

**Seleção de técnicas de geração automática de dados
de teste por meio de aprendizado de máquina**

Versão original

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em xx de xxxxxxxxxxxxxxxx de xxxx. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Prof. Dr. Marcelo Medeiros Eler

São Paulo

2016

Ficha catalográfica

Errata

Elemento opcional para versão corrigida, depois de depositada.

Dissertação de autoria de Fulano de Tal, sob o título **“Seleção de técnicas de geração automática de dados de teste por meio de aprendizado de máquina”**, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em _____ de _____ de _____ pela comissão julgadora constituída pelos doutores:

Prof. Dr. _____

Presidente

Instituição: _____

Prof. Dr. _____

Instituição: _____

Prof. Dr. _____

Instituição: _____

Prof. Dr. _____

Instituição: _____

Lista de figuras

Lista de algoritmos

Lista de tabelas

Sumário

1	Fundamentação Teórica	9
1.1	Área e Tema	9
2	Motivação	10
3	Lacuna/Problema	11
4	Objetivos	12
4.1	Objetivo Geral	12
4.2	Objetivos Específicos	12
5	Hipóteses	13
6	Justificativa	14
7	Métodologia	15
8	Avaliação	16
9	Limitações	17
10	Contribuições	18
	Referências ¹	19

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Fundamentação Teórica

1.1 Área e Tema

Área: Engenharia de Software

Tema: Geração Automática de Casos de Teste

2 Motivação

Produtos de software de diferentes tamanhos e complexidades são utilizados todos os dias em atividades profissionais ou de entretenimento. Em qualquer circunstância, entretanto, a falta de qualidade dos produtos pode caracterizar uma situação preocupante para seus produtores (HILBURN; TOWHIDNEJAD, 2002) (BINDER, 1994), uma vez que cada vez mais os níveis aceitáveis de qualidade de um software estão aumentando, tanto no que se refere ao seu comportamento observável externamente quanto ao seu processo de desenvolvimento e estrutura interna (NORTHROP, 2006) (BASHIR; BANURI, 2008) (GRAHAM; VEENENDAAL; EVANS, 2008). O teste de software (TAHIR; MACDONELL; BUCHAN, 2014) é a maneira mais popular de verificar se um software atende às especificações descritas e cumpre o papel desejado pelos interessados (SOMMERVILLE et al., 2008). Testes são costumeiramente vistos durante o processo como atividades dispendiosas no ciclo de vida de software [8], requisitando tempo, esforço, planejamento e execução (TAHIR; MACDONELL; BUCHAN, 2014), além de recursos financeiros, como demonstrado pelo estudo de impacto econômico dos testes efetuado pelo NIST (National Institute of Standards and Technology), onde o custo nos Estados Unidos para infraestrutura de testes no ano de 2002 é estimado em um valor entre 22 e 59 bilhões de dólares. Aliados aos altos custos, a complexidade e a criticidade de sistemas orientados a objetos são ampliadas (BINDER, 1994), ou seja, os sistemas tornam-se maiores e mais complexos ao passo que os métodos convencionais de validação levam mais tempo e ficam mais caros (TAHIR; MACDONELL; BUCHAN, 2014) (BINDER, 1994). Com o aumento da complexidade e do tamanho do software, ficou mais evidente a necessidade de que os projetos de software devam possuir características que os tornem mais fáceis de serem testados (SHEPPARD; KAUFMAN, 2001), já que compreender e reduzir o esforço de testes são elementos principais de interesse tanto no ramo acadêmico quanto industrial (TAHIR; MACDONELL; BUCHAN, 2014).

3 Lacuna/Problema

Ao longo dos anos diversas técnicas têm sido propostas para o propósito de de geração automática de dados de teste, estas técnicas abrangem conceitos que vão desde análise de estrutura de código até análise dos modelos de dados utilizados em aplicações, porém a área carece de estudos que evidenciem tanto a priorização de técnicas em detrimento de outros em casos específicos de geração assim como a seleção ou combinação destas técnicas para atingir a geração de melhores casos de teste e assim maior cobertura de código.

O processo de geração automática de dados de teste pode se tornar extenso e custoso visto que atualmente os mesmos contam com geração dos casos de teste, redução e/ou priorização dos casos de teste e por fim, aplicação dos casos de teste selecionados. Pra tal, a área carece de uma alternativa como a seleção do melhor método diretamente sem a intervenção de técnicas para selecionar os melhores casos de teste.

4 Objetivos

Nesta seção serão elicitados os objetivos gerais e específicos do projeto afim de preencher a lacuna já detalhada.

4.1 Objetivo Geral

Construir um modelo capaz, a partir da análise de características do código em estudo, de selecionar a melhor técnica de geração automática de dados de teste para o software em questão afim de atingir maior cobertura de código quando comparado às técnicas de geração automática de dados de teste isoladamente.

4.2 Objetivos Específicos

1. mapear sistematicamente as métricas de testabilidade disponíveis na literatura;
2. demonstrar um padrão nas métricas de testabilidade afim de estabelecer uma ferramenta de seleção integrada.
3. compreender as características de software que influenciam a qualidade de casos de teste gerados por cada técnica.
4. selecionar as melhores técnicas de geração automática de casos de teste para softwares em análise.

5 Hipóteses

Nesta seção serão detalhadas as hipóteses deste projeto, as hipóteses a seguir baseiam-se na seleção de técnicas de geração automática de dados de teste por meio de métricas de testabilidade como alternativa aos métodos tradicionais de geração automática e seleção de casos de teste. São elas:

- hipótese 1 : Métricas de testabilidade de software são os melhores indicadores para seleção de técnicas de geração automática de dados de teste;
- hipótese 2 : É possível escolher a melhor técnica de geração automática de dados de teste baseado somente nas métricas de testabilidade extraídas deste código e assim atingir maior cobertura de código com os casos de teste gerados se comparado com as técnicas de geração automática de dados de teste isoladamente.

6 Justificativa

É possível notar através das hipóteses já elicítadas que se as mesmas forem comprovadas como verdadeiras, as estratégias de geração automática de casos de teste atualmente utilizadas poderão ser otimizadas com o uso do modelo proposto por esta pesquisa. Atualmente, engenheiros de qualidade selecionam uma ou mais técnicas de geração automática de casos de teste, como *Symbolic Execution*, *Model-based generation*, entre outras para geração dos casos de teste primários.

A partir dos casos de teste gerados por estas técnicas, o engenheiro de qualidade pode utilizá-los diretamente, porem o número de casos de teste gerados pode ser elevado, consumindo assim mais tempo do que o necessário para validação. Outra opção disponível é selecionar manualmente os melhores casos de teste dentre os gerados para que então sejam utilizados com ferramentas de avaliação adequadas, o problema nesta abordagem consiste primariamente em consumir um tempo elevado para uma tarefa manual e suscetível a erros por parte do engenheiro de qualidade.

Para trabalhar com o possível volume excessivo de casos de teste gerados por diversas técnicas, o engenheiro pode utilizar-se de duas ferramentas que podem ser combinadas como: Redução (ou seleção) e Prioritização de casos de teste. Estas permitem gerar um subconjunto de casos de teste a partir do conjunto inicial, por fim, este subconjunto é colocado a prova pelo engenheiro de qualidade.

Este processo consiste basicamente em quatro etapas: Escolha da técnica de geração automatizada sem base científica para tal; Geração dos casos de teste; redução e/ou prioritização dos casos de teste e por fim, aplicação dos casos de teste. Pretende-se com o modelo em questão reduzir estas etapas a somente duas: Execução do modelo, gerando assim casos de teste que possam servir melhor ao software em análise e por fim aplicação dos casos de teste como no caso anterior.

7 Metodologia

O método deste projeto consiste basicamente em extrair os dados de projetos, treinar o modelo em questão de acordo com os estudos efetuados e por fim treinar o modelo que selecione as técnicas de geração de dados de teste contra cada técnica individualmente.

- Fazer um mapeamento sistemático das métricas de testabilidade existentes na literatura.
- Selecionar e coletar previamente um número reduzido de métricas afim de efetuar testes primários.
- Construir uma rede neural e treina-la a partir do número reduzido de métricas escolhido com o objetivo de validar a convergência da mesma e assim garantir um padrão, validando assim a continuação do projeto.
- Selecionar as melhores métricas ,a partir do mapeamento sistemático , baseado tanto no uso quanto na sua relevância para seleção, caracteriza-se relevância como o quanto uma métrica de software pode descrever uma característica para seleção.
- Selecionar softwares candidatos à análise, a partir de softwares open-source disponibilizados por meio de repositórios de código aberto, tais como: GitHub, Bitbucket, entre outros
- Extrair as métricas selecionadas dos softwares selecionados para análise
- Treinar um modelo afim de selecionar a melhor técnica de geração de dados de teste, baseado nas características do software em questão, características estas representadas por métricas de testabilidade
- Gerar os casos de teste a partir do modelo para cada um dos softwares candidatos à análise
- Gerar os casos de teste para cada um dos softwares candidatos à análise em cada uma das técnicas de geração automática de dados de teste e também para um modelo de seleção randômica de técnicas de geração automática de dados de teste
- Analisar a cobertura de código pontuada por cada um dos conjuntos de casos de teste gerados
- Testar a hipótese de maior cobertura de código gerada pelo modelo contra as estratégias de geração de casos de teste isoladamente

8 Avaliação

Para validar as hipóteses já citadas, é necessário propor um método capaz de avaliar se os casos de teste gerados pelo modelo realmente são melhores se comparados aos casos de teste gerados pelas técnicas de geração de dados de teste isoladamente. Dado que a função principal do processo de teste de software é evidenciar a existência de possíveis defeitos no processo de desenvolvimento de software levando a um erro na execução do mesmo, proporcionando assim a evidência de falhas prematuramente nos processo de desenvolvimento de software, é necessário mensurar e demonstrar a eficiência dos casos de teste gerados pelo modelo em tais condições.

Para obter uma comparação positiva, é necessário que os casos de teste gerados possuam níveis aceitáveis de qualidade, ou seja, os casos de teste gerados automaticamente pelo modelo devem ser possuir níveis melhores de qualidade se comparado às técnicas de geração de dados de teste isoladamente e se comparado a um modelo aleatório de seleção das técnicas de geração de dados de teste. E por fim, para avaliar se a hipótese é aceita, os casos de teste gerados passarão por meio de uma análise de geração de cobertura de código, ou seja, cada conjunto de casos de teste gerado tanto pelo modelo quanto pelas alternativas concorrentes receberá uma porcentagem de cobertura de código para cada um dos softwares analisados.

Espera-se assim que o modelo proposto possua maior cobertura de código na maior parte dos testes executados, superando assim não somente cada técnica de geração automática de dados de teste individualmente como também um modelo de seleção aleatório de técnicas de geração de dados de teste.

9 Limitações

Este trabalho caracteriza-se principalmente com a seleção de técnicas de geração automática de dados de teste a partir de características inerentes de softwares. Sendo assim, o trabalho em questão não pretende focar no contexto de seleção e priorização de casos de teste ou quaisquer contextos que não sejam os relacionados à seleção de técnicas de geração automática de casos de teste, os quais não utilizem métricas de testabilidade como parâmetros de seleção. Salienta-se também que outros métodos que não seja aprendizado de máquina não serão contemplados para seleção das técnicas de geração automática de casos de teste em questão.

Partindo desta premissa, é possível assumir que os seguintes itens não serão contemplados por este projeto, caracterizando-se assim como limitações do mesmo. são elas:

- seleção de casos de teste gerados;
- priorização de casos de teste gerados;
- seleção de técnicas de geração automática de teste por outros métodos que não sejam baseados em métricas de testabilidade;
- automatização em processos de testes funcionais.

10 Contribuições

Nota-se a partir das lacunas já citadas, o potencial proposto pela seleção de estratégias de geração automática de dados de teste quando comparado às técnicas de seleção de casos de teste, sendo assim, este modelo poderá a vir substituir técnicas de seleção de casos de teste muito utilizadas e presentes no estado da arte no que se trata de geração automática de dados de teste.

Espera-se a partir do resultado desta pesquisa, contribuir com um novo modelo para geração automática de dados de teste capaz de gerar automaticamente dados de teste sem necessidade de utilizar técnicas de priorização de dados de teste, consumindo assim menos recursos e atingindo possível cobertura de dados de teste se comparado aos modelos atuais do estado da arte. Espera-se também contribuir para o estado da arte de geração automática de dados de teste, provendo informações no que se trata de melhores características de software para cada técnica de geração de dados de teste, contribuindo assim também para compreender vantagens e desvantagens de cada técnica de geração de dados de teste individualmente.

Referências¹

- BASHIR, M. F.; BANURI, S. H. K. Automated model based software test data generation system. In: *Emerging Technologies, 2008. ICET 2008. 4th International Conference on*. [S.l.: s.n.], 2008. p. 275–279. Citado na página 10.
- BINDER, R. V. Design for testability in object-oriented systems. *Commun. ACM*, ACM, New York, NY, USA, v. 37, n. 9, p. 87–101, set. 1994. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/182987.184077>. Citado na página 10.
- GRAHAM, D.; VEENENDAAL, E. V.; EVANS, I. *Foundations of Software Testing: ISTQB Certification*. Cengage Learning, 2008. ISBN 9781844809899. Disponível em: <https://books.google.ie/books?id=Ss62LSqCa1MC>. Citado na página 10.
- HILBURN, Y.; TOWHIDNEJAD, M. Software quality across the curriculum. In: IEEE. *Frontiers in Education, 2002. FIE 2002. 32nd Annual*. [S.l.], 2002. v. 3, p. S1G–18. Citado na página 10.
- NORTHROP, L. M. Let's teach architecting high quality software. In: *19th Conference on Software Engineering Education and Training (CSEET 2006), 19-21 April 2006, Turtle Bay, Hawaii, USA*. [s.n.], 2006. p. 5. Disponível em: <http://dx.doi.org/10.1109/CSEET.2006.23>. Citado na página 10.
- SHEPPARD, J. W.; KAUFMAN, M. Formal specification of testability metrics in ieeep1522. In: *AUTOTESTCON Proceedings, 2001. IEEE Systems Readiness Technology Conference*. [S.l.: s.n.], 2001. p. 71–82. ISSN 1080-7225. Citado na página 10.
- SOMMERVILLE, I. et al. *Engenharia de software*. ADDISON WESLEY BRA, 2008. ISBN 9788588639287. Disponível em: <https://books.google.com.br/books?id=iflYOgAACAAJ>. Citado na página 10.
- TAHIR, A.; MACDONELL, S. G.; BUCHAN, J. Understanding class-level testability through dynamic analysis. In: *Evaluation of Novel Approaches to Software Engineering (ENASE), 2014 International Conference on*. [S.l.: s.n.], 2014. p. 1–10. Citado na página 10.

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.