

Universidade Federal do Ceará
Pró-Reitoria de Pesquisa e Pós-Graduação
PIBIC 2020/2021 - Edital Nº 1/2020

**Catálogo de Requisitos Não-Funcionais para Sistemas baseados em
Aprendizado de Máquina**

Resumo

Sistemas baseados em aprendizado de máquina são sistemas que utilizam dados para gerar conhecimento e, assim, tomar decisões de forma eficiente. A natureza desses sistemas difere daqueles tradicionais pois são treinados e não explicitamente programados. Recentemente, muita atenção tem sido dada à qualidade desse tipo de sistema, principalmente no que diz respeito aos Requisitos Não-Funcionais (RNFs), como por exemplo, Segurança, Privacidade E Transparência. RNFs são bem compreendidos quando estamos tratando de sistemas típicos. Porém, para um sistema baseado em aprendizado de máquina, esse conhecimento pode não ser mais aplicado. Alguns RNFs podem ter importância reduzida nesses sistemas quando comparados com softwares tradicionais. Por outro lado, outros RNFs podem ganhar importância muito maior. Além disso, RNFs ainda não explorados também podem se tornar relevantes e novas correlações podem surgir. Assim, esse projeto tem como propósito investigar sistemas baseados em aprendizado de máquina existentes no que diz respeito aos RNFs, de forma que seja possível desenvolver um catálogo de RNFs específicos para apoiar desenvolvedores e pesquisadores desse tipo de sistema. Essa investigação será realizada através da execução de métodos de pesquisa e técnicas da Engenharia de Software, tais como Revisões Sistemáticas, Grounded Theory, Content Analysis, uso do Softgoal Interdependency Graph e Experimentos Controlados.

1. Introdução

O Aprendizado de Máquina - do inglês Machine Learning - evoluiu como um subcampo de Inteligência Artificial (IA). Este subcampo oferece uma solução mais eficiente para capturar o conhecimento a partir de dados e assim melhorar gradualmente o desempenho de modelos preditivos e tomar decisões orientadas por dados, em vez de exigir que os seres humanos derivem regras manualmente e construam modelos a partir da análise de grandes quantidades de dados (RASCHKA, S., & MIRJALILI, V., 2019).

O aprendizado de máquina não apenas se torna cada vez mais importante para a pesquisa em Ciência da computação, mas também desempenha um papel cada vez maior nas vidas cotidianas das pessoas. Como exemplos de sistemas baseados em aprendizado de máquina, podemos citar os filtros robustos de spam de e-mail, softwares de reconhecimento de texto e voz, programas de xadrez e, em breve, carros autônomos seguros e eficientes (RASCHKA, S., & MIRJALILI, V. 2019).

Segundo (CHOLLET, F. 2017), o aprendizado de máquina surge das seguintes perguntas: %u201Cum computador poderia ir além do que sabemos como ordená-lo para executar e aprender por si próprio como executar uma tarefa?%u201D e %u201CEm vez de programadores elaborando regras de processamento de dados manualmente, um computador poderia aprender essas regras automaticamente olhando para os dados?%u201D. Assim, um novo paradigma de programação surge. Se antes, os sistemas eram programados de forma que os programadores criam as regras,

nos sistemas baseados em aprendizado de máquina, as regras serão derivadas a partir de dados.

Um sistema baseado em aprendizado de máquina é treinado e não explicitamente programado. Ele é apresentado com muitos exemplos relevantes para uma tarefa e encontra uma estrutura estatística nesses exemplos que eventualmente permite que o sistema crie regras para automatizar a tarefa. No exemplo dado por (CHOLLET, F. 2017), se você deseja automatizar a tarefa de classificar suas fotos de férias, poderá utilizar um sistema baseado em aprendizado de máquina com muitos exemplos de fotos já marcadas por seres humanos, e o sistema aprenderá regras estatísticas para associar fotos específicas a tags específicas.

Em um recente artigo, Horkoff, J. (2019) afirma que muita atenção tem sido dada a certas qualidades das soluções de machine de learning, particularmente Fairness, Transparência, Segurança, Privacidade e Testabilidade. Essas qualidades se referem aos Requisitos Não-Funcionais (RNFs).

Wieggers, K., & Beatty, J. (2013) declaram que existem três tipos de RNFs: (i) características que o sistema deve apresentar (por exemplo, Desempenho), (ii) restrições e (iii) requisitos de interface externa. Este projeto concentra-se em RNFs que são características de qualidade pois estas representam uma porção significativa de RNFs. Essas características descrevem as expectativas além do correto funcionamento do sistema. Por exemplo, expectativas como a rapidez do sistema (Desempenho), quão seguro é o sistema (Segurança) e a facilidade de uso (Usabilidade) são exemplos de características de qualidade.

De acordo com (SILVA et al, 2016), quando as características da qualidade não são atendidas, todo o sistema pode fracassar. Portanto, RNFs do tipo característica de qualidade são igualmente importantes como requisitos funcionais, que são aqueles relacionados às funcionalidades do sistema.

É comum avaliar o sistema em relação a essas características, usando abordagens como Teste de Usabilidade (BEZERRA et al., 2014) e Avaliação de Desempenho (MAIA et al., 2014). No entanto, é necessário e essencial considerar as características de qualidade desde o início do ciclo de desenvolvimento do sistema, pois elas podem afetar a escolha de recursos de hardware e rede, o design da arquitetura e outras questões (SILVA et al, 2016). Depois que o sistema é desenvolvido, é caro ou até impraticável alterá-lo para considerar as características de qualidade (WIEGERS, K., & BEATTY, J. 2013).

Porém, lidar com RNFs do tipo características de qualidade durante o desenvolvimento de software é um desafio. Um dos motivos é que existem muitos tipos deles, e cada um requer conhecimento especializado sobre como apoiá-lo. Pode haver várias soluções alternativas para ajudar uma única característica de qualidade em uma aplicação específica. Esse conhecimento nem sempre está disponível sem esforço para engenheiros de software, especialmente aqueles que não têm experiência no domínio do sistema que está sendo desenvolvido (CHUNG et al., 2000). Outro motivo é que as características de qualidade podem interagir umas com as outras (WIEGERS; BEATTY, 2013), o que significa que atingir uma característica pode impactar o alcance de outra. Esse impacto pode ser positivo ou negativo. Quando negativo, esse impacto é chamado de "conflito" ou "correlação negativa" por (CHUNG et al., 2000). Por exemplo, existem conflitos conhecidos entre Segurança (autorização) e Usabilidade (eficiência) (MAIRIZA et al., 2013). De fato, a característica Usabilidade é impactada negativamente por várias outras características tradicionais de qualidade, como Eficiência, Integridade, Desempenho, Portabilidade e Segurança, sendo uma das NFR mais impactadas (WIEGERS; BEATTY, 2013). Por outro lado, existem correlações positivas, chamadas de "cooperação". Por exemplo, Reutilização ajuda a alcançar Interoperabilidade (WIEGERS; BEATTY, 2013).

Dessa forma, considerando sua importância para o sucesso do software e seus desafios, as características de qualidade têm sido amplamente estudadas e consideradas para o desenvolvimento. Uma das soluções existentes na literatura para ajudar os engenheiros de software a cumprirem características de qualidade é o uso de catálogos de RNFs (CHUNG et al., 2000). Um catálogo é um conjunto de conhecimentos acumulados com experiências prévias. Eles são conhecidos por trazer melhores especificações de requisitos, pois permitem a reutilização de requisitos (CYSNEIROS, 2007) (CARDOSO; GUIZZARDI, 2011) (GRAMATICA et al., 2015). Além disso, o uso de catálogos impede que os engenheiros gastem tempo pesquisando em diversas fontes ou contando com especialistas da área para tomar decisões sobre como obter características de qualidade. Conseqüentemente, eles podem diminuir o tempo de busca e o custo para consultoria de especialistas (CHUNG et al., 2000).

Por exemplo, a literatura possui vários catálogos de RNFs para sistemas tradicionais. Existe catálogo para Segurança, Desempenho e Acurácia em (CHUNG et al., 2000), Transparência em (LEITE; CAPPELLI, 2010), Privacidade em (ZINOVATNA; CYSNEIROS, 2015), Tolerância a Falhas em (TORRES; MARTINS, 2014). Esses catálogos existentes concentram-se em requisitos e estratégias genéricas para qualquer sistema. No entanto, catálogos para sistemas específicos também estão sendo propostos.

Recentemente, estudos indicaram o surgimento de novas características de qualidade de sistemas para o ambiente da Computação Ubíqua e Internet das Coisas (CARVALHO et al., 2016) (ANDRADE et al, 2017), tais como Sensibilidade ao Contexto, Calmness, Invisibilidade, Mobilidade, Atenção e Sincronicidade. Em particular, Invisibilidade foi catalogada (CARVALHO et al., 2018a) utilizando-se um processo chamado CORRELATE proposto na tese de (CARVALHO, 2019). O trabalho descrito na tese consistiu em criar orientações para apoiar pesquisadores e desenvolvedores a refinar e correlacionar RNFs de sistemas ubíquos (CARVALHO, 2017) (CARVALHO et al, 2018b) (CARVALHO et al, 2018c).

Segundo (HORKOFF, J. 2019), se tratando de softwares tradicionais ou domínios mais conhecidos, o significado de certas características de qualidade, como refiná-las e como usá-las para o projeto e execução é relativamente bem entendido, como apresentado acima nos estudos para Computação Ubíqua e Internet das Coisas. No entanto, quando a solução de software envolve aprendizado de máquina, parte do conhecimento existente sobre RNFs pode não se aplicar mais. A maneira como projetamos, executamos, avaliamos e mantemos sistemas baseados em aprendizado de máquina difere devido à natureza desse tipo de sistema, que não são explicitamente programáveis, como afirma (CHOLLET, F. 2017).

Esse questionamento se trata de como podemos adaptar as técnicas e metodologias da Engenharia de Software para sistemas baseados em aprendizado de máquina. Estudos recentes já vêm investigando essa questão (HESENIUS et al., 2019) (AMERSHI et al, 2019) e iniciativas como a SEMLConf (HORKOFF, J. 2019) também já vêm discutindo o assunto. Esse projeto foca em um ponto específico da Engenharia de Software que são os Requisitos Não Funcionais do tipo Característica de Qualidade. Através dele, espera-se encontrar e aprofundar evidências sobre o estado atual dos RNFs em sistemas baseados em aprendizado de máquina e, assim, impulsionar diversas pesquisas futuras para a Engenharia de Requisitos.

2. Perguntas de Partida

A natureza dos sistemas baseados em aprendizado de máquina pode trazer um novo significado para muitos RNFs, já que tais sistemas diferem em comparação com o software comum. Algumas perguntas que surgem são:

- O que significa Manutenção em sistemas baseados em aprendizado de máquina?
- RNFs, como Compatibilidade e Modularidade, ainda são relevantes para esses sistemas?
- A Usabilidade necessitará ser repensada para esse tipo de sistema?
- Existem novos RNFs para esse tipo de sistema? Se sim, quais são eles?
- Existem correlações (conflitos e cooperações) desconhecidas para esses sistemas?

Alguns RNFs podem ter importância reduzida em sistemas baseados em aprendizado de máquina quando comparados com softwares típicos. Por outro lado, outros RNFs podem ganhar importância muito maior. Além disso, RNFs ainda não explorados também podem se tornar relevantes e novas correlações podem surgir.

3. Hipóteses

Neste projeto, argumentamos que RNFs existentes precisam ser investigados e revistos para sistemas baseados em aprendizagem de máquina e que novos RNFs, aqueles que não estão listados em normas padronizadas de RNFs, podem surgir. Dessa forma, nossas hipóteses são:

- Muitos dos RNFs ainda podem ser válidos, porém, eles precisam ser reformulados à luz desse novo paradigma.
- Novas correlações entre RNFs existem em sistemas baseados em aprendizado de máquina.

4. Objetivos

Objetivo Geral

Desenvolver um catálogo de RNFs e suas correlações para sistemas baseados em aprendizado de máquina.

Objetivos Específicos

1. Identificar RNFs relevantes para sistemas baseados em aprendizado de máquina
2. Refinar os RNFs identificados
3. Identificar correlações entre os RNFs identificados
4. Catalogar o conhecimento adquirido
5. Avaliar o uso do catálogo no projeto e desenvolvimento de sistemas baseados em aprendizado de máquina

5. Materiais e Métodos

Para alcançar o objetivo geral do projeto, cada objetivo específico será apoiado através de uma metodologia de pesquisa e/ou técnicas bem conhecidas da Engenharia de Software.

Para o primeiro objetivo específico, o método a ser utilizado é a revisão sistemática da literatura (KITCHENHAM, 2007) e o Survey (WOHLIN et al., 2012). Revisão Sistemática é uma metodologia a partir de evidências baseadas em engenharia que serão usadas para apoiar a revisão da literatura sobre o assunto em relação a um objetivo. Em geral, esse método consiste em três fases: planejamento da revisão, que estabelece o protocolo com as questões de pesquisa de revisão a serem respondidas, bem como a estratégia de busca de artigos; realização da revisão, que é a pesquisa, análise e avaliação dos trabalhos em relação ao nosso objetivo; e disseminação dos resultados, onde os resultados são documentados e divulgados aos interessados. Como resultado, definimos um protocolo de pesquisa com perguntas de pesquisa, palavras-chave, sequências de pesquisa, critérios de inclusão/exclusão, informações de extração de dados. A análise estatística dos resultados (aplicável ou não) é produzida, sendo definida uma lista final dos elementos que respondem às perguntas da pesquisa.

Survey é um sistema para coletar informações de ou sobre pessoas para descrever, comparar ou explicar seus conhecimentos, atitudes e comportamento. (WOHLIN et al. 2012) Uma survey geralmente é uma investigação realizada em retrospecto, quando, por exemplo, uma ferramenta ou técnica está em uso há um tempo. O principal meio de coleta de dados qualitativos ou quantitativos do survey são entrevistas ou questionários. Isso é feito através da coleta de uma amostra representativa da população a ser estudada. Os resultados da pesquisa são então analisados para derivar conclusões descritivas e explicativas. Eles são generalizados para a população da qual a amostra foi coletada. Dessa forma, um survey deverá ser executado com especialistas em sistemas baseados em aprendizado de máquina.

Para o segundo e terceiro objetivo específico, um processo específico para refinar e definir correlações entre RNFs (CARVALHO 2019) será utilizado. Em poucas palavras, esse processo pressupõe que dois requisitos não funcionais estejam correlacionados devido às estratégias de desenvolvimento usadas pelos desenvolvedores para construir o sistema. Portanto, é necessário entender primeiro quais são as estratégias utilizadas para atingir uma característica e depois investigar como essas estratégias afetam outras características de qualidade. Assim, foi criado um processo com quatro etapas e técnicas de apoio para correlacionar dois requisitos não funcionais. Essas técnicas são baseadas nos métodos Grounded Theory e Content Analysis (CHO, J. Y. & LEE, E.-H. 2014).

Ainda para o terceiro objetivo, experimentos controlados com sistemas existentes serão executados. Experimento em engenharia de software é uma investigação empírica que manipula um fator ou variável (WOHLIN et al. 2012). Com base na randomização, diferentes tratamentos são aplicados por diferentes sujeitos, mantendo outras variáveis constantes e medindo os efeitos nas variáveis de resultado. Para esse projeto, há a possibilidade de investigar correlações através da manipulação de RNFs, assim como em (FEITOSA, 2015).

Para o quarto objetivo, a notação Softgoal Interdependency Graph (CHUNG et al., 2000) deverá ser utilizada. Nessa notação, a característica de qualidade é tratada como um softgoal, que é uma meta sem critérios claros de satisfação. Nesta notação, os softgoals devem ser refinados em objetivos secundários (isto é, subcaracterísticas), os quais devem ser refinados em operacionalizadores (isto é, soluções de design de implementadores para suportar objetivos) e, finalmente, esses operacionalizadores podem apresentar correlações em relação a outros softgoals. Este trabalho utilizará essa notação para o catálogo de RNFs de sistemas baseados em aprendizado de máquina devido à sua ampla utilização na literatura. Além disso, esse tipo de catálogo permite analisar em profundidade como diferentes características de qualidade podem ser satisfeitas com estratégias de implementação.

Para alcançar o quinto objetivo, experimentos controlados e estudos de casos deverão ser utilizados (WOHLIN et al. 2012). Experimentos, descritos anteriormente, podem ser executados para verificar a eficiência e eficácia de um catálogo no desenvolvimento desse tipo de sistema em um ambiente controlado. Por outro lado, estudo de caso é uma investigação empírica que recorre a várias fontes de evidência para investigar uma instância de um fenômeno de engenharia de software em seu contexto da vida real, especialmente quando a fronteira entre fenômeno e contexto não pode seja claramente específico. Dessa forma, estudos de caso em um contexto real, ou seja, da indústria, deverá ser executado.

6. Referências Bibliográficas

AMERSHI, S., BEGEL, A., BIRD, C., DELINE, R., GALL, H., KAMAR, E., ... & ZIMMERMANN, T. Software engineering for machine learning: A case study. In 2019 IEEE/ACM 41st International

Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 291-300). IEEE, 2019.

ANDRADE, R. M. C.; CARVALHO, R. M.; ARAUJO, I. L.; OLIVEIRA, K. M.; MAIA, M. E. F.; What changes from Ubiquitous Computing to Internet of Things in Interaction Evaluation? In: International Conference on Human-Computer Interaction (HCI), 2017.

BEZERRA, C. I. M. et al. Challenges for Usability Testing in Ubiquitous System. In: *Interaction Homme-Machine*. [S.l.: s.n.], 2014.

CARVALHO, R. M.; ANDRADE, R. M. C. ; OLIVEIRA, K. M.; SANTOS, I. S; BEZERRA, C. I. M. Quality characteristics and measures for human-computer interaction evaluation in ubiquitous systems. *Software Quality Journal* 2016

CARVALHO, Rainara Maia; ANDRADE, Rossana; DE OLIVEIRA, Káthia Marçal. Correlations between invisibility and usability in ubicomp and IoT applications: partial results. In: *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. ACM, 2018. p. 214-219. c

CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M.; KOLSKI, C. Catalog of Invisibility Requirements for UbiComp and IoT Applications. In: *IEEE International Requirements Engineering Conference* , 2018a.

CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M. Towards a Catalog of Conflicts for HCI Quality Characteristics in UbiComp and IoT Applications: Process and First Results. In: *IEEE 12th International Conference on Research Challenges in Information Science*, 2018.b

CARVALHO, R. M. Dealing with Conflicts between Non-Functional Requirements of Ubicomp and IoT Applications In: *Doctoral Symposium at IEEE International Requirements Engineering Conference* , 2017.

CARVALHO, Rainara Maia. CORRELATE & LEAD: process and catalog of non-functional requirements correlations in ubicomp and iot systems. 2019. 230 f. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Ceará, Fortaleza, 2019.

CHO, J. Y.; LEE, E.-H. Reducing confusion about grounded theory and qualitative content analysis: Similarities and differences. *The qualitative report*, v. 19, n. 32, p. 1-20, 2014.

CHOLLET, F. *Deep Learning with Python*. 978 1617294433. Shelter Island: Manning Publications, 384. 2017.

CHUNG, L.; NIXON, B. A.; YU, E.; MYLOPOULOS, J. *Non-functional requirements in software engineering*. [S.l.]: Springer Science and Business Media, 2000. v. 5

LEITE, J. C. S. d. P.; CAPPELLI, C. *Software transparency*. *Business and Information Systems Engineering*, Springer, v. 2, n. 3, p. 127-139, 2010.

FEITOSA D, AMPATZOGLOU A, AVGERIOU P, NAKAGAWA EY. Investigating quality trade-offs in open source critical embedded systems. In *2015 11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA) 2015 May 4* (pp. 113-122). IEEE.

HESENIUS, M., SCHWENZFEIER, N., MEYER, O., KOOP, W., & GRUHN, V. Towards a software engineering process for developing data-driven applications. In *2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)* (pp. 35-41).

IEEE. 2019.

HORKOFF, J. Non-functional requirements for machine learning: Challenges and new directions. In 2019 IEEE 27th International Requirements Engineering Conference (RE) (pp. 386-391). IEEE. 2019.

MAIA, M. E.; ROCHA, L. S.; ANDRADE, R. M. Requirements and challenges for buildingservice-oriented pervasive middleware. International conference on Pervasive services -ICPS, ACM Press, 2009.

MAIA, M. E. F. et al. USABLE - A communication framework for ubiquitous systems. Proceedings - International Conference on Advanced Information Networking and Applications, AINA, n. 481417, p. 81%u201388, 2014.

KITCHENHAM, B. Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE-2007-01. Department of Computer Science Keele University, Keele, 2007.

RASCHKA, S., & MIRJALILI, V. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd. 2019

SILVA, A.; PINHEIRO, P.; ALBUQUERQUE, A.; BARROSO, J. A Process for Creating the Elicitation Guide of Non-functional Requirements. In: Software Engineering Perspectives and Application in Intelligent Systems. [S.l.]: Springer, 2016. p. 293%u2013302

WIEGERS, K., & BEATTY, J. Software requirements. Pearson Education. 2013.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. Experimentation in software engineering. [S.l.]: Springer Science and Business Media, 2012.

7. Plano de Atividades

Mês	Bolsista 1	Bolsista 2
1	Estudo Inicial e Planejamento da Revisão Sistemática	Estudo Inicial e Planejamento do Survey
2	Execução da Revisão Sistemática na Literatura	Execução do Survey com Especialistas
3	Análise e Relato dos Resultados da Revisão	Análise e Relato dos Resultados do Survey
4	Agrupamento e Catalogação dos RNFs da Revisão e do Survey	Agrupamento e Catalogação dos RNFs da Revisão e do Survey
5	Seleção de um RNF para refinamento e Execução do CORRELATE	Planejamento do Experimento de Correlações com Sistemas Existentes
6	Execução do CORRELATE (continuação)	Execução do Experimento de Correlações para Análise de Correlações
7	Execução do CORRELATE (continuação)	Execução do Experimento de Correlações para Análise de Correlações (continuação)
8	Execução do CORRELATE (continuação)	Análise e Relato dos Resultados dos Experimentos com Sistemas Existentes

9	Planejamento do Estudo de Caso na Indústria	Catálogo das Correlações em SIG e planejamento de experimento do uso do catálogo
10	Execução do Estudo de Caso na Indústria	Execução da Avaliação do Catálogo por meio de Experimento
11	Análise e Relato de Resultados	Análise e Relato dos Resultados
12	Escrita de Artigo	Escrita de Artigo