



CENTRO UNIVERSITÁRIO CTC DEPARTAMENTO DE INFORMÁTICA		Ano e semestre 2023.2
INF2005	TÓP. ENGENHARIA DE SOFTWARE I (Replicabilidade, Reprodutibilidade e Reusabilidade)	
Dia: 6ª feira	CARGA HORÁRIA TOTAL: 45h	CRÉDITOS: 3
Horário: 13 - 15 + 1 SHF	Pré-requisitos: Não há Prof. Juliana Alves Pereira	

<b>OBJETIVOS</b>	<p>Soluções de aprendizagem de máquina possibilitam a automatização de diversas atividades do processo de desenvolvimento de software – tais como a detecção e correção de defeitos de software. Como o desenvolvimento de soluções de aprendizagem de máquina podem envolver altos custos, elas devem ser projetadas de tal forma a serem facilmente reutilizáveis em diferentes aplicações. Entretanto, pouco se sabe empiricamente se e quando as soluções existentes são reutilizáveis. O objetivo desta disciplina é, através da proposição e condução de replicações de estudos, discutir e compreender os diversos desafios com relação a replicabilidade destas soluções.</p>
<b>EMENTA</b>	<p>Tópicos abordados: Conceitos fundamentais sobre o uso de soluções de aprendizado de máquina para engenharia de software. Seminários sobre o uso de soluções de aprendizado de máquina em outras áreas além da engenharia de software (conforme interesses dos alunos matriculados). Introdução a boas práticas de projeto associadas ao desenvolvimento dessas soluções. Boas práticas de replicabilidade, reprodutibilidade e reusabilidade. Técnicas e ferramentas para replicabilidade. Formação de grupos de trabalho e seleção de estudos para replicação. Condução da replicação de estudos. Introdução ao Qualis. Introdução sobre o processo de revisão em pares e submissão de trabalhos para periódicos e eventos científicos. Introdução a escrita de artigos. Boas práticas de apresentação em conferência. Revisão em pares dos trabalhos produzidos na disciplina. Apresentação final.</p> <p><b>Importante:</b> o programa sofrerá ajustes conforme as expectativas e interesses dos alunos matriculados na disciplina.</p>

<p><b>PROGRAMA</b></p>	<p>A disciplina será composta essencialmente por uma série de mesas redondas para discussões e replicações de artigos recentes na área.</p> <p><b>Fase 1: Introdução da área</b></p> <p>Esta primeira fase é voltada para alunos novos que não tenham conhecimento prévio na área. Portanto, esta fase inicial será dividida em duas partes: (i) apresentação da área (aprendizado de maquina para engenharia de software e outras disciplinas), e (ii) introdução dos três “R”s de <i>Open Science</i>: Replicabilidade, Reprodutibilidade e Reusabilidade.</p> <p><b>Fase 2: Formação de grupos de trabalho e seleção do estudo replicável</b></p> <p>Nesta fase, os alunos serão alocados em grupos de acordo com seu tema de interesse. Cada aluno deve selecionar um artigo potencial para replicação.</p> <p>Pontos a serem considerados:</p> <ul style="list-style-type: none"> <li>• Problema abordado</li> <li>• Motivação da replicação</li> <li>• Detalhamento da metodologia usada pelo artigo original</li> <li>• Proposta de metodologia a ser seguida pelo grupo para a replicação do estudo selecionado <ul style="list-style-type: none"> <li>○ questões de pesquisa</li> <li>○ metodologia aplicada</li> <li>○ esforço necessário para replicação</li> <li>○ contribuições esperadas</li> </ul> </li> <li>• Detalhamento da base de dados original disponibilizada pelos autores</li> <li>• Seleção de um evento de alto rigor científico para submissão do trabalho</li> </ul> <p><b>Fase 3: Acompanhamento das tarefas de replicação do estudo</b></p> <p>Nessa fase, após selecionado o artigo alvo de replicação, serão discutidas:</p> <ul style="list-style-type: none"> <li>• Questões de pesquisa</li> <li>• Metodologia e cronograma</li> <li>• Desenvolvimento das tarefas (coleta de dados para reuso, replicação do artigo original, desenvolvimento de melhorias, condução de experimentos, etc.)</li> <li>• Resultados</li> </ul> <p>Para cada um dos itens entregáveis, o grupo deve apresentar ao professor o estágio atual do trabalho, de forma que os demais colegas possam discutir a solução do problema e propor melhorias. Nestas aulas os alunos devem expor</p>
------------------------	--

	<p>o andamento desde a proposta realizada, informando quaisquer desvios em relação ao planejado. O objetivo desta fase é acompanhar o progresso do trabalho até a submissão do trabalho final.</p> <p><b>Fase 4: Apresentações, revisões em pares e submissão do trabalho</b></p> <p>Esta fase corresponde a apresentação final do trabalho com forte enfoque nos resultados e contribuições obtidas e na submissão do mesmo. Estas apresentações serão agendadas de acordo com deadline dos eventos escolhidos e devem ter o formato equivalente a apresentações feitas em conferências. Antes das apresentações, serão realizadas revisões alternadas em pares dos trabalhos pelos colegas e professora para prover um feedback completo aos grupos durante a apresentação e submissão.</p>
<p><b>AVALIAÇÃO</b></p>	<p>Atividades de acompanhamento + Trabalho final + Revisão em pares + Apresentações</p>
<p><b>BIBLIOGRAFIA PRINCIPAL</b></p>	<p>Abou Khalil, Z., &amp; Zacchiroli, S. (2022, September). Software Artifact Mining in Software Engineering Conferences: A Meta-Analysis. In ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2022).</p> <p>Acher, M., Martin, H., Pereira, J. A., Lesoil, L., Blouin, A., Jézéquel, J. M., &amp; Barais, O. (2022, September). Feature Subset Selection for Learning Huge Configuration Spaces: The case of Linux Kernel Size. In Systems and Software Product Line Conference (SPLC 2022).</p> <p>Amann, S., Beyer, S., Kevic, K., &amp; Gall, H. (2013). Software mining studies: Goals, approaches, artifacts, and replicability. In Software Engineering (pp. 121-158). Springer, Cham.</p> <p>Brooks, A., Roper, M., Wood, M., Daly, J., &amp; Miller, J. (2008). Replication's role in software engineering. In Guide to advanced empirical software engineering (pp. 365-379). Springer, London.</p> <p>Herbold, S., Trautsch, F., Harms, P., Herbold, V., &amp; Grabowski, J. (2019). Experiences with replicable experiments and replication kits for software engineering research. In Advances in Computers (Vol. 113, pp. 315-344). Elsevier.</p> <p>Lewowski, T., &amp; Madeyski, L. (2022). How far are we from reproducible research on code smell detection? A systematic literature review. Information and Software Technology, 144, 106783.</p> <p>Liu, C., Gao, C., Xia, X., Lo, D., Grundy, J., &amp; Yang, X. (2021). On the reproducibility and replicability of deep learning in software engineering. ACM Transactions on Software Engineering and Methodology (TOSEM), 31(1), 1-46.</p>

	<p>Mahmood, Z., Bowes, D., Hall, T., Lane, P. C., &amp; Petrić, J. (2018). Reproducibility and replicability of software defect prediction studies. <i>Information and Software Technology</i>, 99, 148-163.</p> <p>Martin, H., Acher, M., Lesoil, L., Jezequel, J. M., Khelladi, D. E., &amp; Pereira, J. A. (2021). Transfer learning across variants and versions: The case of linux kernel size. <i>IEEE Transactions on Software Engineering</i>.</p> <p>Martin, H., Acher, M., Pereira, J. A., &amp; Jézéquel, J. M. (2021, September). A comparison of performance specialization learning for configurable systems. In <i>Proceedings of the 25th ACM International Systems and Software Product Line Conference-Volume A</i> (pp. 46-57).</p> <p>Pereira, J. A., Acher, M., Martin, H., Jézéquel, J. M., Botterweck, G., &amp; Ventresque, A. (2021). Learning software configuration spaces: A systematic literature review. <i>Journal of Systems and Software</i>, 182, 111044.</p> <p>Pereira, J. A., Acher, M., Martin, H., &amp; Jézéquel, J. M. (2020, April). Sampling effect on performance prediction of configurable systems: A case study. In <i>Proceedings of the ACM/SPEC International Conference on Performance Engineering</i> (pp. 277-288).</p> <p>Pineau, Joelle, et al. Improving reproducibility in machine learning research. <i>Journal of Machine Learning Research</i> 22 (2021).</p> <p>Uchôa, A., Barbosa, C., Coutinho, D., Oizumi, W., Assunção, W. K., Vergilio, S. R., ... &amp; Garcia, A. (2021, May). Predicting design impactful changes in modern code review: A large-scale empirical study. In <i>2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)</i> (pp. 471-482). IEEE.</p> <p>Wang, S., Huang, L., Ge, J., Zhang, T., Feng, H., Li, M., ... &amp; Ng, V. (2020). Synergy between machine/deep learning and software engineering: How far are we?. <i>arXiv preprint arXiv:2008.05515</i>.</p> <p>Watson, C., Cooper, N., Palacio, D. N., Moran, K., &amp; Poshyvanyk, D. (2022). A Systematic Literature Review on the Use of Deep Learning in Software Engineering Research. <i>ACM Transactions on Software Engineering and Methodology (TOSEM)</i>, 31(2), 1-58.</p>
<p><b>BIBLIOGRAFIA COMPLEMENTAR</b></p>	