


TOP ENGENHARIA DE SOFTWARE I (Soluções de Aprendizagem de Máquina para Engenharia de Software)		
CENTRO UNIVERSITÁRIO CTC DEPARTAMENTO DE INFORMÁTICA		
		Ano e semestre 2024.1
INF2006	Prof ^a . Juliana Alves Pereira	
Dia: 6 ^a feira	CARGA HORÁRIA TOTAL: 45H	CRÉDITOS: 3
Horário: 13 – 15h +1SHF	PRÉ-REQUISITOS: Não há	

OBJETIVOS	Esta disciplina tem como principal objetivo explorar e aplicar técnicas avançadas de aprendizado de máquina no contexto específico da engenharia de software. Essa abordagem visa integrar métodos e algoritmos de aprendizado de máquina para otimizar processos, melhorar a qualidade e eficiência no desenvolvimento de software, bem como para proporcionar insights valiosos para os profissionais da área.
EMENTA	<p>Tópicos abordados: Introdução a engenharia de software. Conceitos fundamentais sobre o uso de soluções de aprendizado de máquina para engenharia de software. Introdução a boas práticas de projeto associadas ao desenvolvimento dessas soluções. Boas práticas de replicabilidade, reprodutibilidade e reusabilidade. Introdução a escrita e submissão de artigos para periódicos e eventos científicos. Introdução sobre o processo de revisão em pares. Boas práticas de apresentação em conferência. Seleção de estudos para replicação e formação de grupos de trabalho. Condução da replicação de estudos. Revisão em pares dos trabalhos produzidos na disciplina. Apresentação final.</p> <p>Importante: o programa sofrerá ajustes conforme as expectativas e interesses dos alunos matriculados na disciplina.</p>
PROGRAMA	<p>Fase 1: Introdução da área</p> <p>Esta primeira fase é voltada para alunos novos que não tenham conhecimento prévio na área. Portanto, esta fase inicial será dividida em três partes: (i) Fundamentos de Engenharia de Software, (ii) Aprendizado de Máquina para Engenharia de Software, e (iii) Introdução dos três “R”s de Open Science: Replicabilidade, Reprodutibilidade e Reusabilidade. Por fim, iremos discutir como esses princípios científicos básicos são essenciais para replicação de</p>

experimentos em engenharia de software com a apresentação detalhada de um conjunto de artigos recentes na área que seguem tais princípios.

Fase 2: Introdução a publicação científica

O objetivo nessa fase é preparar os alunos para a produção de trabalhos acadêmicos e científicos de qualidade. Nessa fase, os alunos são guiados através dos elementos fundamentais da escrita científica, com ênfase na clareza, precisão e conformidade com as normas acadêmicas. O objetivo principal é capacitar os alunos a comunicar de maneira eficaz suas pesquisas e ideias, seguindo os padrões reconhecidos pela comunidade científica. Além disso, busca-se entender todo o ciclo de vida do processo de publicação, desde a escrita até a submissão e revisão, finalizando com a apresentação em conferências científicas. Essa abordagem visa proporcionar ao aluno uma compreensão abrangente do ambiente acadêmico, incentivando a participação ativa em diferentes etapas do processo de pesquisa e comunicação científica. Ao final desta fase, os alunos estarão melhor preparados para enfrentar os desafios e expectativas do meio acadêmico, contribuindo significativamente para o avanço do conhecimento em suas respectivas áreas de estudo.

Fase 3: Formação de grupos de trabalho e seleção do estudo replicável

Nesta fase, os alunos serão alocados em grupos de acordo com seu tema de interesse. Cada aluno deve selecionar um artigo potencial para replicação. Essa fase será composta por uma série de mesas redondas para discussões desses artigos.

Fase 4: Detalhamento do estudo e acompanhamento das tarefas

Nessa fase, após selecionado o estudo alvo de replicação, os demais integrantes do grupo devem realizar sua leitura completa. Após leitura, através de mesas redondas, o grupo realiza a:

- definição das questões de pesquisa
- elaboração da metodologia
- distribuição de tarefas (coleta de dados para reuso, desenvolvimento de um novo algoritmo, melhoria de um script existente, condução de experimentos, etc.)
- definição de cronograma
- discussão dos resultados
- escrita do artigo

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 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>Fase 5: Apresentações, revisões em pares e submissão do trabalho 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 antes da submissão.</p>
<p>AVALIAÇÃO</p>	<p>Atividades de acompanhamento + Revisão em pares + Trabalho final</p>
<p>BIBLIOGRAFIA PRINCIPAL</p>	<p>Abou Khalil, Z., & 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., & 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., & 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., & 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., & 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., & 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., & 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., & Petrić, J. (2018). Reproducibility and replicability of software defect prediction studies. Information and Software Technology, 99, 148-163.</p>

	<p>Martin, H., Acher, M., Lesoil, L., Jezequel, J. M., Khelladi, D. E., & 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., & 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., & 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., & 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., ... & 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., ... & 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., & 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>BIBLIOGRAFIA COMPLEMENTAR</p>	