LASSE F. WOLFF ANTHONY

Machine Learning Engineer ◇ Danish National
+45 81 19 71 00 ◇ lfwa@proton.me ◇ lfwa.github.io ◇ github.com/lfwa ◇ linkedin.com/in/lfwa

EDUCATION

ETH Zürich 2020 – 2023
MSc ETH in Computer Science *Zürich, Switzerland*
Major in Machine Intelligence and minor in Programming Languages and Software Engineering.
Thesis: "Exploring Data Collection Dynamics Through Data Valuation."

University of Copenhagen 2017 – 2020
BSc in Computer Science *Copenhagen, Denmark*
Specialization in Data Science with coursework focused on machine learning.
Thesis: "The Carbon Footprint of Training Deep Learning Models."

University of California, Merced 2019 – 2019
UCEAP Reciprocal Program *Merced, California, USA*
Exchange study part of my BSc CS at UCPH with coursework focused on machine learning.

EXPERIENCE

UBS 2023 – Present
Quant Developer *Zürich, Switzerland*

· Design and develop big data tools and solutions for Treasury Risk Control's balance sheet analytics.

Alexandra Institute 2023 – 2023
AI / Machine Learning Specialist *Copenhagen, Denmark*

· Dual role in applied research and expert consultancy in machine learning at a Research and Technology Organization with a focus on natural language processing (NLP) and utilizing pretrained transformers.

University of Copenhagen 2020 – 2020
Teaching Assistant *Copenhagen, Denmark*

· TA for the course Data Science (2019/2020): Databases, introduction to machine learning and data science with an emphasis on natural language processing and data pipelines.

Nykredit 2018 – 2020
Software Developer *Copenhagen, Denmark*

· Worked as a software developer in an agile C# development team, utilizing Scrum and Kanban methodologies at one of Denmark's leading financial institutions.
· Developed and maintained financial software for mortgage loans, significantly reducing time spent on processing loan application for financial advisors.
· Implemented a continuous deployment pipeline using a Jenkins CI server integrated with BitBucket, fully automating integration testing and deployment.

PUBLICATIONS

[1] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," in *ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems*, Jul. 2020.
[2] R. Selvan, N. Bhagwat, L. F. W. Anthony, B. Kanding, and E. B. Dam, "Carbon footprint of selecting and training deep learning models for medical image analysis," in *International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI 2022*, 2022.

HIGHLIGHTED PROJECTS

Carbontracker
`github.com/lfwa/carbontracker`
Open-source tool for tracking and predicting the energy consumption and carbon emissions of training deep learning models in Python. The tool is freely distributed under the MIT License. Corresponding publication [1]. It has been downloaded >75k times on the Python Package Index (PyPI) as of writing.

Datadynamics
`github.com/lfwa/datadynamics`
Open-source library and environment for simulating data collection dynamics in multi-agent settings, primarily targeting the exploration of data valuation approaches. The library is freely distributed under the BSD 3-Clause License.

Reinforced Graph Neural Networks for Collaborative Filtering
`github.com/lfwa/reinforced-gnn`
Introduced a novel architecture to generate predictive compatibility scores for never-before-seen content in recommendation systems. The architecture combines the strength of graph-extracted embeddings in a graph neural network with the generalization power of a deep feed-forward network and adds "reinforcements" providing additional information to the network.

Static Taint Analysis For Ethereum Contracts
`github.com/lfwa/vulnerable-ethereum-contracts`
Designed and implemented a static taint analyzer in Datalog for Ethereum smart contracts. The analyzer detects vulnerable contracts that may be deleted from the blockchain and have all remaining cryptocurrency transferred to an untrusted address.

Supporting Alternative SMT Solvers in Viper
`github.com/viperproject`
Added support for multiple SMT solvers, such as cvc5, in the symbolic-execution based automated verification backend written in Scala for the program verification tool chain and infrastructure, Viper.

RELEVANT COURSEWORK

Machine Learning & Big Data	**Mathematics**	**Software Engineering**
Advanced Machine Learning	Statistics & Probability Theory	Program Verification
Causal Representation Learning	Discrete Mathematics	Program Analysis for System
Natural Language Processing	Linear Algebra	↳ Security and Reliability
Probabilistic AI	Modelling & Analysis of Data	Concepts of Object-Oriented
Reliable & Trustworthy AI	Algorithms & Data Structures	↳ Programming
Computational Intelligence	Randomized Algorithms	Computer & Network Security
Big Data		

SKILLS

Programming Languages	Python, C#, SQL, Rust, Scala, F#, Java, C, Datalog
Databases	PostgreSQL, Oracle
Frameworks and Tools	PyTorch, TensorFlow, Gym(nasium), PettingZoo, NumPy, pandas, scikit-learn, Matplotlib, Git, Spark, Hadoop, Neo4j, QuantLib