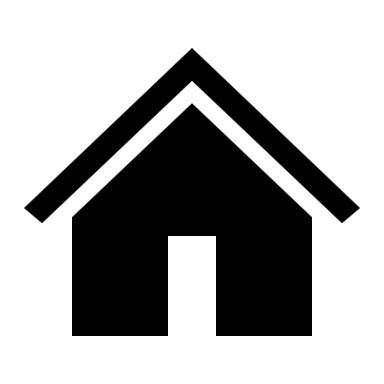
Yu-Sheng (Adam) Tang

Experienced Data Engineer for 5 years, Freiburg im Breisgau, Germany

 [+49 1578-129-1537](tel://+4915781291537/) | [ adam951502@gmail.com](mailto:adam951502@gmail.com) | [ *adam951502*](https://github.com/adam951502) | [ *yu-sheng-tang*](https://linkedin.com/in/yu-sheng-tang) | [ *adam951502.github.io*](https://adam951502.github.io/)

# Education

## Graduation capUniversity of Freiburg MarkerGermany Oct 2018 - Feb 2021

Master of Science in Sustainable Systems Engineering German Grade: 1.7, GPA: 3.6/4.0

## Graduation capChang Gung University MarkerTaiwan Sep 2012 - Jun 2016

Bachelor of Science in Mechanical Engineering German Grade: 1.3, GPA: 3.8/4.0

\* Dean’s list Award: Feb 2014, Sep 2014, Feb 2015

# 

# Skills

**Languages**: Python, MATLAB, C++, Bash, LaTeX

**DBMS**: PostgreSQL, MySQL, Ontotext GraphDB (SPARQL)

**Web**: HTML/CSS, JavaScript, Bootstrap, Flask, RESTful APIs, Swagger, JWT

**Data Skills**: ETL, data wrangling, web scraper, web crawling, data warehousing, data catalog

**Tools & Techs**: pypi, pip, git, pgadmin4, International Data Space

**DevOps**: CI/CD (GitHub, GitLab), AWS, Jenkins, Pytest, Docker, uWSGI, NGINX

**LLMs & RAG:** Llama3, Ollama, Open WebUI, LangChain

**Management**: Jira & Confluence (admin, maintainer), project management, agile management, requirement engineering

**Certificate**: IBM Data Science Professional Certificate,  
International Requirement Engineering

# Experience

## Briefcase Data Engineer & Data Scientist | Building[Fraunhofer EMI |MarkerGermany](https://www.emi.fraunhofer.de/en/business-units/automotive/digital-engineering.html) Oct 2021 - Present

**Technical Skills:**

* Designed an ontology for over 3,000 datasets and built a knowledge graph (KG) for material life cycle assessment (LCA) of additive manufacturing processes using Python and SPARQL, enhancing data pipeline, analysis, and query capabilities by 50%.
* Designed architecture and use-case of resilience data space.
* Constructed an efficient ETL data pipeline in Python, significantly enhancing efficiency by 80%.
* Established a Python package and RESTful APIs (Flask) to streamline data processing, enhancing productivity and code reusability by 90%.
* Established microservices backend APIs using Flask, uWSGI, NGINX, and Docker, reducing deployment time by 80%.
* Implemented DevOps and CI/CD pipeline for automated testing, release, and deployment, enhancing code quality and project efficiency in GitLab by 40%.
* Deployed and integrated large language models (LLMs), Llama 3, using Ollama, Open WebUI, and Docker to enhance the (meta)data pipeline, increasing processing speed by 70%.
* Established a complete ETL pipeline for the public health database using PostgreSQL, pgAdmin4, GraphDB, Flask, uWSGI, NGINX, and Docker, improving data accessibility by 80%.

**Project Management:**

* Supervised a master's student, providing guidance throughout the course of the project (with German Grade: 1.0) – *Designing An Automated Metadata Extraction Pipeline: Bridging the Gap Between Data Collection and Unified Graph-Based Metadata Representation*, resulting in a 85% completion rate.
* Supervised 2 students in establishing an ETL pipeline for a comprehensive metadata graph that describes heterogeneous data from various sources, achieving a 90% increase in data processing efficiency.
* Developed a universal use-case guideline and template, significantly enhancing the efficiency and quality of software development processes within the HERAKLION project, leading to a 30% improvement in project delivery times.
* Managed requirements (requirement engineering), agile project management cycle, and project execution for a team of more than 20 people using Jira, streamlining workflows and improving efficiency by 35%.

# Projects

## [HERAKLION](https://www.emi.fraunhofer.de/en/business-units/automotive/digital-engineering.html) *- Heuristic Resilience Analyses for Municipalities Using Data Space Functionalities* Mar 2022 - Present

***Tech Stack: pip, git, CI/CD, DevOps, ETL, RESTFul API, Python, py-library, Flask, Docker, KG, requirement engineering, agile management***

* Aimed to enhance crisis data accessibility and usability for municipalities and emergency forces by creating a data 'ecosystem.' Developed a scalable resilience data space demonstrator, addressing user needs and ensuring all stakeholders access to critical information for effective and timely crisis responses. The model is intended for nationwide use in Germany.

## ADAM-SusTrace *- Networking of digital assets and data-driven value creation through data ecosystems in additive manufacturing* Oct 2021 - Feb 2022

***Tech Stack: git, Python, algorithm, data structure, Ontology, knowledge graph, LCA***

* Applied digital traceability in additive manufacturing to conduct comprehensive sustainability analyses (Life Cycle Assessment, LCA). Assisted in developing tools and workflows, enhancing data ecosystem's value, and facilitating detailed understanding of product/process sustainability.

## XOR Neural Networks Apr 2022 - Apr 2022

***Tech Stack: Python, NNs***

* This project involves the development of a single-layer perceptron and a multi-layer perceptron (MLP) to understand and predict XOR logic. Both models are trained on the training data and tested on the test data. The performance and decision boundaries of the models can be visualized.

## AHP Operation Research Oct 2019 - Feb 2020

***Tech Stack: Python, Numpy, xlrd, Multi-criteria decision making, TOPSIS, WSM, AHP***

* Implemented the Analytical Hierarchy Process (AHP) and other decision-making methodologies (WSM, TOPSIS) in Python to analyze and rank electricity generation technologies based on various sustainability criteria for optimal decision-making in operations research.

## Renewable Energy Market Value Analysis Apr 2019 - Jul 2019

***Tech Stack: Python, NumPy, Pandas, Matplotlib, Scikit-learn, Energy market value***

* A model was created to analyze the market value of wind and solar power in various electricity markets, including Germany, France, and Sweden. Python was used to develop the model and visualize the generated data.