Muhammad Shariq Khan

M.Eng. AI Engineering for Autonomous Systems

engr.m.shariqkhan@gmail.com +491782141562

Erkner, Brandenberg, Germany linkedin.com/in/muhammadshariqkhan/

github.com/muk0644

ABOUT ME

AI Engineer with experience in LLMs and Generative AI, currently pursuing a Master's in AI Engineering for Autonomous Systems. Possessing expertise in Deep Learning, Machine Learning, Computer Vision, Data Engineering and analytics, and Python-based AI automation. Actively exploring LLMs and transformer-based models for autonomous systems. Passionate about applying AI solutions and contributing to innovative organizations that value critical thinking, collaboration, and continuous learning.

EDUCATION AND TRAINING

Master of Engineering (M.Eng.) AI Engineering of Autonomous Systems 01/10/2023 – Present
Technische Hochschule Ingolstadt Ingolstadt, Germany

Bachelor of Engineering (B.E) Mechanical 13/10/2017 – 14/10/2021
NED University of Engineering and Technology Karachi, Pakistan
Grade: 1.3 (German grade)

WORK EXPERIENCE

Working Student | LLM for Traffic Perception 01/10/2025 – 31/01/2026
CARISSMA – Institute of Automated Driving (C-IAD) Ingolstadt, Germany

- Fine-tuned the multimodal LLM Qwen2-VL-7B using LoRA (PEFT) on the large-scale UrbanIng-V2X dataset, boosting counting accuracy from **14%** (zero-shot) to **93%** and reducing Mean Absolute Error (MAE) from 2.77 to 0.17 per frame.
- Engineered a zero-shot inference pipeline using Qwen2-VL-7B with Hugging Face Transformers to establish performance baselines for scene understanding.
- Processed high-definition autonomous driving datasets (ROS2 bag/.db3) using Python, SQLite3, and Linux pipelines to extract frames and stitch videos for the UrbanIng-V2X workflow.
- Conducted a literature study on modern transformer-based VLMs (e.g., CLIP, BLIP, LLaVA) to evaluate their relevance for cooperative perception tasks.
- Derived video-encoder embeddings for latent representation analysis and executed video clustering including latency optimization.
- Tech Stack: PyTorch, Transformers, PEFT/LoRA, NumPy, Git, and Bash in GPU-accelerated workflows on an Ubuntu Linux workstation.

Working Student | Plant Digitalization & Data Management 08/2024 – 07/2025
Gunvor Raffinerie Ingolstadt GmbH Ingolstadt, Germany

Developed data extraction and transformation pipelines using SQL and Python for daily operational analytics, alongside establishing automated KPI dashboards in Power BI (HSSE, Production) that improved reporting efficiency by 40%. Furthermore, these automated systems and statistical analysis of time-series data reduced manual reporting by 30% and delivered actionable production insights.

Trainee Engineer | Predictive Maintenance & Data Analytics 02/2022 – 10/2023
National Refinery Limited Karachi, Pakistan

Optimized reliability and Condition-Based Maintenance (CBM) for critical rotary equipments by analyzing noisy sensor data, utilizing Python (Pandas) and Power BI to boost data accuracy by 25% through improved cleaning/visualization, resulting in reduced equipment downtime and better maintenance scheduling.

TECHNICAL SKILLS

AI & Machine Learning
- PyTorch
- TensorFlow / Keras
- LangChain / LlamaIndex
- Transformers (Hugging Face)
- PEFT / LoRA
- Large Language Models (LLMs)
- Scikit-Learn
- OpenCV
- Reinforcement Learning (Q-Learning, DQN)

Data Processing, Evaluation & Visualization
- NumPy
- Pandas
- Data Preparation & Cleaning
- Model Training & Evaluation
- Matplotlib
- Seaborn
- Microsoft Power BI
- ROS2 bag / .db3 handling
- Sensor Data Fusion

Programming & Tools
- Python
- Git, GitHub
- Docker
- Linux / Bash
- FastAPI
- MySQL
- Kubernetes
- MongoDB
- AWS
- CI/CD

LANGUAGE SKILLS

English – C1 (Fluent) ● ● ● ● ● German – C1 (Fluent) ● ● ● ● ●

PROJECTS

SmolAgents AI Assistant: Agentic RAG & MLOps Pipeline 10/2025 – 01/2026
Technische Hochschule Ingolstadt

- Built a multi-model AI agent using Hugging Face smolagents, supporting Qwen2.5-Coder-32B and Gemini 2.5 Flash for autonomous task execution and complex tool-calling.
- Architected an Agentic RAG system by integrating LangChain for document chunking, implementing a semantic knowledge base via BM25, and utilizing Chroma DB for high-precision retrieval.
- Integrated custom toolkits for real-time capabilities, including SerpAPI and DuckDuckGo for web research, OpenWeatherMap, and an ephemeral text-to-image generation pipeline.
- Designed a robust LLMOps pipeline via GitHub Actions for systematized CI/CD and Docker, incorporating mechanized security scans (Bandit/Safety) and continuous deployment.
- Launched a production-ready interface on HuggingFace Spaces and Streamlit Cloud, featuring a secure, email-based multi-session chat history system.

Urban Traffic Analysis with Deep Learning 03/2025 – 07/2025
Technische Hochschule Ingolstadt

- Deep Learning Model: Developed and trained a custom PyTorch Neural Network to predict missing Traffic Occupancy, achieving a 61% reduction in Mean Squared Error (from 2.24 to 0.87) over 100 epochs.
- Data Pipeline: Handled and cleaned multi-gigabyte datasets (millions of records across ~1,000 detection points) using Pandas/NumPy to establish a robust predictive data pipeline.
- ML Pattern Recognition: Applied K-Means Clustering for unsupervised pattern recognition, identifying and validating 7 distinct urban traffic flow behaviors (e.g., congestion vs. free-flow) using the Elbow Method.
- Spatial Analysis: Leveraged Geospatial Analysis (Folium, Euclidean Distance) to automatically detect faulty sensors and map their 5 nearest neighbors for precise data imputation.

Reinforcement Learning : Autonomous Agent Training using Q-Learning & DQN 03/2024 – 08/2024
Technische Hochschule Ingolstadt

- Constructed a custom Reinforcement Learning (RL) environment using Python, Gymnasium, and Pygame for dynamic simulation training.
- Implemented core RL techniques, including Q-Learning and a Deep Q-Network (DQN) with PyTorch, for neural network-based value-function approximation.
- Built and managed essential RL components like Q-networks, epsilon-greedy exploration, and reward shaping mechanisms.
- Tuned hyperparameters to achieve stable policy convergence across training iterations, effectively balancing exploration and exploitation.
- Visualized training performance using Matplotlib/Seaborn and performed model optimization via the Adam optimizer.