Nigo Elnagar

Machine learning R&D @ Siemens AG

www.nigo.dev contact@nigo.dev Berlin, Germany Nigo Elnagar

Professional Experience

Siemens AG, Machine Learning R&D — Jul 2023 – present | Berlin

ML Researcher, AI R&D Division: Conduct research on deep learning architectures for computer vision-based detection and semantic interpretation of electrical schematics and symbols. Lead comparative analysis of state-of-the-art object detection frameworks (**YOLO** variants, **Co-DETR**, **InternImage-H**, **Faster R-CNN**), segmentation models (**SAM 2**), vision transformers (**Dual Attention ViT**, **DINOv2**), and multimodal architectures (**Qwen2.5-VL**) powered by distributed training pipelines on **Azure ML Studio**, achieving **>96%** mean average precision across multiple model configurations and test datasets.

Deutsche Bank, Machine Learning - Technology, Data & Innovation (TDI) division — Jan 2023 – Jun 2023 | Berlin

Developed a **holistic multi-agent** forecasting framework integrating news sentiment with multivariate time series data, leveraging Hybrid **GARCH-CNN-LSTM** models for multiscale volatility decomposition, **attention-augmented CNN-BiLSTM** with self-adaptive optimization for non-stationary series, **Channel-independent Patch Time Series Transformer** employing self-attention for high-dimensional non-linear dependencies, and **DLinear** for efficient, low-latency inference in high-frequency trading environments.

Projects & Competitions

BMW Agents - Multi-Agent Collaboration Framework For Task Automation., GitHub

Implementing hierarchical task decomposition with deterministic DAG-based execution. Features bi-directional agent communication, vector-embedded episodic memory, semantic toolbox refinement, and configurable prompt strategies. Enables autonomous collaborative workflows through specialized agent persona instantiation and adaptive matching algorithms.

Memory-augmented Agentic Information Retrieval, GitHub

Implementation of Zhang et al. Agentic IR paradigm with a memory-augmented agent architecture. Features stateful information transitions, thought generation, policy learning, and tool integration powered by local LLMs. A modular framework for research on multi-step, reasoning-based information retrieval.

Vertical Agents Implementation, GitHub

Agentic system with BaseMemory, ShortTermMemory, LongTermMemory, VectorMemory. Human-Augmented Agents and RAG Router for knowledge management. Vector Embeddings for semantic search and In-Memory Vector Database. LLM Integrations (via Ollama, deepseek-r1) and Async Communication Protocols.

Intel ISEF, Dec 2015 – May 2016

Recognized as one of the world's premier pre-college science competitions, I presented an electrostatic engine tailored for electric vehicles at this esteemed event and was honored with a **bronze medal** for my contribution.

Transformer-based News Summarization BART, GitHub

BART transformer for extractive news summarization, demonstrating significant convergence with cross-entropy loss reduction from **1.5276** to **0.1102** between initial training epochs. Quantitative evaluation via **ROUGE** metrics indicated robust performance (**rouge1=0.7753**, **rouge2=0.6970**, **rougeL=0.6110**, **rougeLsum=0.6119**), optimized to **147.54** seconds per batch inference. Incorporated **Weights & Biases** for parameter tracking, with the resultant model architecture published to the **Hugging Face Model Repository** for reproducibility and deployment.

Linked-based Classification using Graph Neural Networks, GitHub

A Graph Convolutional Network was implemented for link prediction on the Cora citation network. Through systematic evaluation of diverse train-validation-test partitioning protocols, an optimal data stratification ratio of **0.4:0.1:0.5** was identified, yielding a minimized cross-entropy validation loss of **1.6321** and maximum test accuracy of **87.89%** upon convergence (epoch 27). Incorporated Batch Virtual Adversarial Training (BVAT)

Physics Informed Neural Networks, GitHub
Implemented a Physics-Informed Neural Network in PyTorch for 1D harmonic oscillators, integrating an underdamped oscillator's analytical solution. Employed a unique loss function combining data fidelity and physical law compliance, with iterative visualizations of model training.

Prophet-based Time Series Forecasting of Twitter Stock Data, GitHub
Leveraged Facebook's Prophet library for Twitter stock forecasting, employing advanced trend analysis, seasonality decomposition, and changepoint detection techniques. Model accuracy was quantified with Mean Absolute Error, complemented by Plotly visualizations for actual vs. predicted value comparison, showcasing predictive efficacy.

Pedestrian Detection using Histogram of Oriented Gradients, GitHub
Developed a pedestrian detection system using OpenCV's Histogram of Oriented Gradients (HOG) in Python. Applied Sobel operators for gradient calculations in Cartesian coordinates, converted to polar for magnitude and orientation analysis, and visualized using gradient direction quivers and weighted HOG histograms. Demonstrated sophisticated feature extraction and visualization techniques for effective object recognition.

Face Detection and Recognition, GitHub
Developed a real-time face detection and recognition system using OpenCV and face_recognition in Python. Integrated with facial encodings stored using pickle, the system identifies faces in video frames, matches them with known encodings, and displays names, exemplifying advanced real-time biometric identification techniques.

Forecasting Web Traffic using supervised ML Algorithms (Multimodel-Analysis), GitHub
The project adeptly applied supervised machine learning algorithms to predict web traffic, achieving notable **accuracies of 81.07%** with **Logistic Regression and SVM**. Precise data preprocessing involved outlier detection using **Mahalanobis distance** measurements for **multivariate anomaly identification**, **custom-defined invalid rows detection**, and **categorical binning** for **variable discretization**. Strategic **Hyperparameter tuning** enhanced model efficacy, as evidenced by Logistic Regression and SVM's superior **ROC AUC scores of 0.87** and **minimal log loss**, reflecting high predictive reliability and discriminative precision within the digital marketing domain.

Optimizing non-convex functions using Particle Swarm Optimizer, GitHub
A cutting-edge implementation of Particle Swarm Optimization (PSO) tailored for navigating and optimizing complex non-convex functions. This project encapsulates an advanced algorithmic approach, leveraging swarm intelligence to efficiently converge on global minima in multimodal landscapes.

Implementing ML Algorithms from Scratch, GitHub
- Architectured and implemented a comprehensive library of foundational machine learning algorithms in Python, showcasing a deep understanding of ML concepts and applications.
- Developed algorithms include **AdaBoost**, **Decision Trees**, **k-Nearest Neighbors (KNN)**, **Linear and Logistic Regression**, **Naive Bayes**, **Principal Component Analysis (PCA)**, **Random Forest**, and **Support Vector Machines (SVM)**.
- Accompanied each model with **rigorous testing** via Jupyter notebooks, displaying thorough validation and performance analysis.

Quantum KNN Classifier using Qiskit, GitHub
The implementation involves constructing a quantum circuit with multiple registers: **index**, **training**, and **test quantum registers**, alongside an **auxiliary qubit** for similarity measurement. Data encoding is achieved through **amplitude encoding**, converting classical data points into quantum states. A pivotal aspect of the project is the integration of a **custom Quantum SWAP Test module**, crucial for calculating the similarity between encoded quantum states. The quantum circuit is measured using a classical register, and the results are processed to predict the class of test samples based on k nearest neighbors in the training set. The model's efficacy is demonstrated through an accuracy assessment against classical test data, underlining the practical applications of quantum algorithms in machine learning.

Evaluate Quantum Fourier Transform using Quantum Machine Learning, GitHub
Implemented Quantum Fourier Transform (QFT) using PennyLane for circuit creation and optimization. Employed RMSProp for parameter optimization in a two-qubit system, achieving precise target state alignment. Visualized optimization process using 3D plots in Matplotlib, demonstrating the convergence of quantum state probabilities.

Using Simulated Annealing for the Traveling Salesman Problem, GitHub
Engineered a Simulated Annealing solution to optimize the Traveling Salesman Problem, effectively minimizing the total route distance by crafting a Python implementation that leverages an adjacency matrix for distance computation. Iterative refinement over 50,000 iterations culminated in a route cost of 28 units, demonstrating the algorithm's efficacy in converging towards an optimal solution. Key techniques included stochastic perturbations, temperature decay functions, and Metropolis acceptance criteria, with results visualized through NetworkX and Matplotlib, substantiating the algorithm's performance in navigating and escaping local optima within the solution landscape.

Studienkolleg T-Kurs Hochschulzugangsberechtigung (FSP), Note: 1,3 — Oct 2020 – Jul 2021

TU München, Computer Science B.Sc. — Oct 2021 – present | München

Einführung in die Informatik | Einführung in die Rechnerarchitektur | Diskrete Strukturen | Grundlagenpraktikum: Programmierung | Einführung in die Softwaretechnik | Grundlagen: Algorithmen und Datenstrukturen | Funktionale Programmierung und Verifikation | Lineare Algebra für Informatik | Grundlagenpraktikum Rechnerarchitektur

Humboldt Universität Berlin, Physics B.Sc. — Oct 2022 – Sep 2023

Linear Algebra | Analysis I | Mechanics | Mathematical Fundamentals and Basic Lab | Analysis II | Theoretical Mechanics

Credentials - Educational Background

AI Agents Using RAG and LangChain	IBM
MLOps Platforms: Amazon SageMaker and Azure ML	Duke University
Computer Vision in Microsoft Azure	Microsoft
Encoder-Decoder Architecture	Google
Machine Learning in Production	DeepLearning.AI
Enterprise Model Deployment	IBM
Large Language Model Operations (LLMOps)	Duke University
ML Pipelines on Google Cloud	Google
Generative AI Engineering and Fine-Tuning Transformers	Duke University
Visual Perception	Columbia University
GenAI and LLMs on AWS	Duke University
Attention Mechanism	Google
Microsoft Azure Machine Learning for Data Scientists	Microsoft
Computer Science and Programming Using Python	MIT
Advanced MySQL Topics	Meta
Generative AI Advance Fine-Tuning for LLMs	IBM
MLOps | Machine Learning Operations	Duke University
Generative AI Language Modeling with Transformers	IBM
TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning	DeepLearning.AI
Operationalizing LLMs on Azure	Duke University
Fundamentals of Red Hat Enterprise Linux	Red Hat
Google Advanced Data Analytics	Google
AI Workflow: Machine Learning, Visual Recognition and NLP	IBM
Databricks to Local LLMs	Duke University

DataCamp

116 Hours | 6 Projects

Database Structures and Management with MySQL — Meta

Open Source LLMOps Solutions — Duke University

Regression Analysis: Simplify Complex Data Relationships — Google

Virtualization, Docker, and Kubernetes for Data Engineering — Duke University

Data Science using Python — Microsoft

MLOps Tools: MLflow and Hugging Face — Duke University

DevOps, DataOps, MLOps — Duke University

Analysis of Algorithms — Princeton University

Coding Interview Preparation — Meta

Create Machine Learning Models in Microsoft Azure — Microsoft

Advanced Understanding of Stocks and Bonds — University of Michigan

Introduction to Quantum Circuits — The Linux Foundation

Advanced Concepts in Time Value of Money (TVM) — University of Michigan

Advanced Data Engineering — Duke University

Stocks and Bonds — University of Michigan

Foundational Finance for Strategic Decision Making — University of Michigan

Java Programming and Software Engineering Fundamentals — Codecademy

√ Skills

Mathematical & Statistical Analysis
Mathematical Modeling | Statistical Analysis & Hypothesis Testing | Analysis of Algorithms | Mathematical Optimization Techniques

Deep Learning Architectures
Mixture-of-Experts Models| Multi-modal Architectures | Convolutional Neural Networks | Diffusion Models | Self-Supervised Learning | Recurrent Neural Networks | Long Short-Term Memory | Physics-Informed Neural Networks | BART Transformer | Graph Convolutional Networks | Gated Recurrent Units | Autoencoders | Variational Autoencoders | Generative Adversarial Networks | Reinforcement Learning | Llama

ML & Data Science Toolkit:
Azure ML Studio | AWS | TensorFlow | PyTorch | Keras | Scikit-Learn | Pandas | NumPy | Matplotlib | Seaborn | Plotly | Tableau | spaCy | NLTK | FastAI

Computer Vision & Image Processing
YOLO models | DETR | SAM 2 | DINOv2 | Qwen 2.5 | Faster R-CNN | Vision Transformers | DenseNet | U-Net | ResNet | Histogram of Oriented Gradients | DeepFace | OpenCV | scikit-image | Inception | DaViT | CoAtNet-7

ML Algorithms
XGBoost | Gradient Boosting Machines | LightGBM | Random Forests | AdaBoost | SVM | Decision Trees | | Naive Bayes Classifiers | Boosting | Bagging | K-Means Clustering | KNN | PCA | Elastic Net Regression | DBSCAN | Linear Regression | Polynomial Regression | Logistic Regression | Lasso Regression

Programming & Software Development
Java | Python | C++ | Rust | Julia | OOP | Shell Scripting (Bash-CLI) | Database Management (MySQL / MongoDB) | Docker | Kubernetes | AWS | Version Control (Git & GitHub) | CI/CD

Languages

English ● ● ● ● ●

German ● ● ● ● ○
C1 Certified