

# Jaganathan Ramkumar

+4917645533812 | [achillesram@gmail.com](mailto:achillesram@gmail.com) | [linkedin.com/in/ramkumar-jaganathan](https://linkedin.com/in/ramkumar-jaganathan) | [github.com/jettyindeep1](https://github.com/jettyindeep1)

## SUMMARY

Applied AI specialist with experience building intelligent systems from structured and unstructured data, taking solutions from concept to deployment. Background spans software development, Machine Learning, computer vision, with a current focus on generative AI based solutions—delivering high-impact, compliant AI solutions.

## EDUCATION

<b>M. Tech in Data Science and Engineering</b> <i>Birla Institute of Technology and Science</i>	2020 – 2022 <i>Work Integrated Learning</i>
<b>B. Engg in Electronics and Communication Engineering</b> <i>Coimbatore Institute of Technology</i>	2002 – 2006 <i>Coimbatore, India</i>

## EXPERIENCE

<b>Lead AI/ML Engineer</b> <i>Bosch Engineering GmbH</i>	Jan. 2024 – Present <i>Holzkirchen, Germany</i>
<ul style="list-style-type: none"><li>Leading technical development of capturing and organizing engineering artefacts together with their relationships as a data backbone to enable <b>GenAI powered agentic applications</b> for bringing productivity improvements in systems engineering processes</li><li>Architected an <b>AI powered pdf-importer application</b> that extracts requirements text and requirement metadata from requirement pdf files and ingests them into a Databricks delta table. Utilised <b>AZURE AI Document Intelligence</b> for OCR and layout extraction, An <b>LLM based pipeline</b> for automotive requirement specific text classification</li><li>Introduced <b>user-in-the-loop for annotating a few shot examples for text classification</b> and using those examples to <b>fine-tune class descriptions with an LLM powered interactive prompt engineering system</b></li><li>Implemented an alternative text classification method using <b>contextual bandits</b> that dynamically adapts to user feedback on classification results to improve classification accuracy over time.</li><li>Implemented an LLM powered Requirements Similarity Assistant performing semantic comparisons of new and legacy requirements to identify similarity and changes using <b>RAG techniques</b> with <b>vector embeddings</b> for retrieval and similarity comparison</li><li>Together with data engineering team, helped design <b>data pipelines and ETL processes</b> to ingest and process large volumes of requirements engineering data and created <b>bronze, silver, and gold delta tables in Databricks</b>.</li></ul>	
<b>CV/ML Engineer</b> <i>Bosch Engineering GmbH</i>	Feb. 2018 – Dec. 2023 <i>Holzkirchen, Germany</i>
<ul style="list-style-type: none"><li>Designed and developed a <b>probabilistic graphical model based generative AI algorithm</b>, predicting evolving height map of dump trucks during excavator loading, resulting in 45% increase in average accuracy using Python, Matlab, and Bayesian methods. The method used LiDAR point cloud and vehicle kinematic data and vehicle signals as inputs to predict height profile of load on truck bed.</li><li>Designed and developed a <b>boltzmann network based deep neural network model</b> to extract and locate partially buried excavator bucket in 3D LiDAR point cloud data and excavator kinematic data.</li><li>Designed and developed a bucket fill level estimation algorithm using geometric segmentation and <b>DBSCAN clustering algorithm</b> to segment 3D LiDAR points belonging to bulk material in excavator bucket and estimate fill level using 3D regression technique.</li><li>Designed and developed a soil pile identification algorithm from 3D LiDAR point cloud data by defining a height threshold and classifying points above and below into a binary image and using <b>connected component analysis</b> to identify soil piles.</li><li>Designed and developed a dump truck identification algorithm from 3D LiDAR point cloud data using <b>RANSAC based plane fitting</b> to identify truck bed plane.</li><li>Designed and developed an algorithm using <b>support vector machines</b> to classify reachability of a 3D point in space by an excavator's tool tip.</li></ul>	

- Designed and developed excavator **kinematics tree** and **inverse kinematics tree** models from joint angle sensor data and LiDAR mounting position using **MATLAB** and **Linear Algebra Methods** and performed coordinate transformations from one frame to another for LiDAR data processing.
- Designed and developed overlay visualisation of Radar and Ultra sound detected obstacles on different view images (Birds eye, rear view, side view) of a surround camera system using **ROS** for receiving sensor data and **OpenCV** for image processing.
- Filled **11 patents** related to the development of the above algorithms and received a **Bosch Inventor Award**.

## Software Engineer

Jun. 2015 – Jan. 2018

*Bosch Engineering K.K*

*Yokohama, Japan*

- Adapted and developed control algorithms for engine exhaust gas treatment systems to meet stringent emission norms using MATLAB/Simulink, ASCET and C
- Automated the process of defining AUTOSAR definitions from network definition files, reducing manual effort by 70% using Python
- Led review process to enhance team performance and software reliability while maintaining traceability across development artifacts

## Software Engineer

Aug. 2006 – Mar. 2015

*Robert Bosch Engineering and Business Solutions Ltd*

*Bangalore, India*

- Developed and adapted bootloader, CAN, Diagnostics and hardware related software modules for automotive ECUs based on AUTOSAR standards using C
- Maintained traceability between software development artifacts to optimize documentation and workflow
- Provided technical leadership during review process and facilitated cross-functional collaboration

## PROJECTS

---

### **BYOS - Build Your Own Survey** | *Python, OpenAI GPT-4, LlamaIndex, Docker*

- Developed an intelligent survey creation agent using OpenAI GPT-4 LLM and LlamaIndex that automatically generates contextual surveys based on user prompts and document attachments
- Implemented multi-agent orchestration system with vector embeddings for document retrieval, automated question generation, and web search integration for research-backed questions

### **BYOC - Build Your Own Classifier** | *Python, OpenAI GPT-4, Few-Shot Learning*

- Developed an interactive few-shot learning system that iteratively refines text classifiers through human-in-the-loop learning
- System uses OpenAI GPT-4 LLM to generate targeted questions about unlabeled text, collects user feedback, and automatically updates class descriptions based on predictions vs. ground truth
- Implemented three-stage pipeline: question generation, interactive annotation, and class refinement, enabling adaptive classification without traditional model retraining

## TECHNICAL SKILLS

---

**Programming Languages:** Python, C, C++, MATLAB

**AI Libraries:** PyTorch, Scikit-learn, Langchain, OpenCV, PCL, LlamaIndex

**Platforms & Tools:** Databricks, PySpark, ROS, Git, Docker, VS Code

**AI Methodologies:** Machine Learning, Deep Learning, Probabilistic Graphical Models, Graph ML, Reinforcement Learning, RAG, Agentic AI

## LANGUAGES

---

**Tamil:** Native | **English:** C2 | **Japanese:** B2 | **German:** A2

## CERTIFICATIONS

---

**Completed:** [Reinforcement Learning](#), [Improving Deep Neural Networks](#), [Convolutional Neural Networks](#), [Neural Networks and Deep Learning](#)

**Ongoing:** AI Governance Professional