

## EDUCATION

### University of Central Florida - Expected Graduation: 2027

B.S. Computer Science | B.S. Statistics | Minor: Actuarial Sciences

#### Relevant Coursework

Calculus With Analytical Geometry II/III • Discrete Structures • Object Orientated Programming • Statistical Methods I/II  
Computer Logic and Organization • Computer Science I/II • Algo for Machine Learning • Statistical Theory I

## SKILLS

**Programming Languages:** Python3 • C • SQL • Java • HTML/CSS • JavaScript

**Frameworks & Tools:** DJANGO Framework • React Native • REST APIs • AWS EC2 • GCP Compute Engine • Flask • Git

**Libraries:** Keras/TensorFlow • MatPlotLib • Pandas Library • Neural Networks • Word2Vec • ReportLabs • OpenCV

## EXPERIENCE

Undergraduate Research Assistant – University of Central Florida | November 2024 – Present

- Collaborated with a project group focused on “Developing Algorithms for Intelligent Autonomous Systems,” working alongside PhD students and faculty principal investigators.
- Advanced project objectives in Multi-Agent Reinforcement Learning (MARL) and distributed optimization through research, algorithm design, and system implementation.
- Developed and compared multiple MARL Policy Evaluation techniques, primarily focusing on the performance of Local Temporal Difference Updates.
- Used the Petting Zoo API to simulate different environments to test the performance of the algorithms, while using performance metrics like Mean Squared Bellman Error and Consensus Error.
- Integrated control theory and machine learning principles to design and implemented control systems for Arduino-controlled robots, contributing to the field of intelligent autonomous systems.

## PROJECTS

Smart home assistant — Jul - Aug 2024

- Developed a Smart Home Assistant using Raspberry Pi, Python, and TensorFlow.
- Implemented facial recognition for secure door unlocking via a servo motor. Automated lighting control using a radio frequency transceiver. Integrated music playback for hands-free entertainment. Set up VPN server functionality to ensure secure and private internet access.
- Utilized TensorFlow, OpenCV, ArduCam, Flask, and MQTT to combine machine learning, hardware integration, and network security.

Compiler for Pseudocode in C — July 2024

- Designed and implemented a complete compiler in C, encompassing lexer, parser, semantic analyzer, and code generator for a bespoke pseudo-code language.
- Ensured modular architecture to allow easy scalability and maintenance, supporting future expansion of language features.
- Integrated comprehensive error handling to provide clear and informative feedback, improving user experience and debugging efficiency.

Trading strategy Algorithm — Jul 2024

- Developed a Python program to analyze stock data using technical indicators like SMAs, EMAs, MACD, RSI, and Bollinger Bands. Pulled stock data via the yfinance library, applied feature engineering, and implemented clustering using Gaussian Mixture Models (GMM) to identify market patterns.
- Generated buy/sell signals and visualized market trends with Matplotlib by plotting technical indicators and entry signals.
- Utilized Random Forest Classifier and Gaussian Mixture models to predict stock movements, with data preprocessing using StandardScaler and feature selection from multiple technical indicators.

HACKATHON – ANATOMY ANT — Lead Backend Development– Oct 2024

- Solely developed the full backend for a mobile app designed to aid Anatomy students. The app incorporated chapter summaries fed through an OpenAI API, chapter quizzes created by ChatGPT and a finetuned Anatomy OpenAI model.
- The Flask backend, hosted on a Google E2 instance, returns JSON data with anatomy questions, including fields for chapter, question ID, text, options, correct answer, and user's choice. Worked with front-end team to integrate APIs into front-end.
- All anatomy-related information was generated using a finetuned OpenAI 4.0 API and fed into the frontend through a Flask API

Embeddings Data Pipeline – Oct 2024

- Developed a FastAPI application to generate and store text embeddings via OpenAI's Embedding API, designing robust RESTful endpoints for embedding creation and retrieval. Utilized Pydantic for rigorous data validation alongside FastAPI to build a reliable and efficient API, ensuring data integrity throughout the service.
- Implemented asynchronous database operations with SQLAlchemy and PostgreSQL, leveraging Alembic for seamless schema migrations to maintain database consistency and performance. Planned future enhancements to evolve the system into a Retrieval-Augmented Generation (RAG)-enabled agent, enabling advanced, context-aware responses and expanded AI capabilities

## CERTIFICATIONS

Learn Python 3 (Code Academy) • Learn SQL (Code Academy) • Learn HTML (Code Academy) • Learn CSS (Code Academy)