Austin Shearin

Email: austinmshearin@gmail.com
LinkedIn: https://www.linkedin.com/in/austin-shearin

Summary
Data scientist/engineer, solutions architect, software developer with leadership and project management experience and an applied physics and materials science education. Technical projects include establishing extract, transform, load processes on-premises and in the cloud; architecting cloud infrastructure to deploy software as a service applications and support IoT sensing systems; developing web-based applications for data management, analysis, and visualization; and utilizing statistical models and machine learning for correlations, modeling, and simulations.

Education

M.S. Materials Science
GPA 3.97
Missouri State University

B.S. Physics
GPA 3.89
Missouri State University

Technical Certifications

AWS Certified Cloud Practitioner, Amazon Web Services
Foundational understanding of AWS Cloud
Python Programmer, DataCamp
Software engineering, unit testing, parallel processing
Data Scientist, Analyst with Python, DataCamp
Exploratory data analysis, hypothesis testing, machine learning, databases
Statistical thinking for industrial problem-solving, SAS
EDA, DOE, quality methods, correlation/regression, statistics
Scrum Project Management, The Mathis Group
Agile SCRUM methodologies

Work Experience

SynTouch, Inc.
Director of Data Science 02/2022 – Present

- Lead the software development, solutions architecture, and data science initiatives within the company
- Project manager for all SaaS application development using Agile SCRUM methodologies in JIRA
- Architected the cloud infrastructure in AWS to support microservice SaaS applications using Route 53, ALB, ECS, Lambda, RDS, Elasticache, Cognito, and Secrets Manager
- Develop single page applications as containerized services in Python using Streamlit, Plotly, SciPy, Scikit-Learn, Pandas, Numpy, PyMySQL, and AWS SDK
- Principal investigator of research projects involving haptic perception (or touch) and haptic preference
- Point of contact for all customer inquiries regarding data analysis and SaaS applications
- Write and deliver technical reports for customers who contract services with the company

Brewer Science
Sr. Device Systems Engineer – Team Leader 09/2021 – 02/2022
Device Systems Engineer – Team Leader 01/2020 – 09/2021
Associate Device Development Engineer 06/2018 – 01/2020

- Lead a team of principal investigators for several IoT sensing system development projects
- Project manager for sensing system development projects using Agile SCRUM methodologies in JIRA
- Principal investigator of an IIoT sensing system that could predict failure modes of industrial equipment
- Architected the cloud infrastructure in AWS to support ETL processes of IoT data packets using IoT Core, Lambda, RDS, Elasticache, and ECS
- Developed ETL processes to combine, clean, aggregate, format, and validate IoT data packets to store in RDS and Elasticache in Python using Numpy, Pandas, PyMySQL, and AWS SDK
- Technical point of contact with customers, beta phase partners, and joint development members

Research Associate III 06/2017 – 06/2018
Research Associate II 05/2016 – 06/2017

- Research associate in the flexible hybrid electronics business unit
- Performed root cause analysis on analog sensors utilizing DOE
- Designed custom test equipment to measure sensor performance to varying stimuli
- GUIs for test equipment were developed in Python using Tkinter, pyserial, matplotlib, Numpy, and Pandas to manually control test equipment or automate test routines

Example Projects

Software as a Service Data Management, Visualization, and Analysis Platform

The web-based data management, visualization, and analysis platform was designed to assist customers when working with data from the equipment the company sold that produced a fifteen-dimensional dataset. It generated a new revenue stream for the company as customers did not want to invest in complex software for data analysis.

I designed the cloud infrastructure to host the application in Amazon Web Services (AWS). The application was containerized and deployed in Elastic Container Service (ECS) utilizing Fargate compute resources behind an Application Load Balancer (ALB). Additional AWS services include Route 53, Cognito, Secrets Manager, Relational Database Service (RDS), Simple Storage Service (S3), Lambda, CloudWatch, and Simple Email Service (SES).

I developed the single page application (SPA) in Python utilizing Streamlit. The application provided services for data management, experimental design, data visualization, statistics tests, and training predictive models. Plotly was used to generate interactive figures to display multi-dimensional data. Scikit-learn was used to perform data preprocessing, feature engineering, and dimensionality reduction; investigate correlations between features; and train, optimize, and compare different machine learning models. Other packages used to support the application worth mentioning were Pandas, NumPy, SciPy, PyMySQL, and the Python AWS SDK.

Industrial Internet of Things Sensing System

The industrial internet of things (IIoT) sensing system was designed to predict when failure modes occurred on a piece of industrial equipment. The system recorded high frequency time series measurements when events were detected and transmitted the data over cellular to the cloud where it was analyzed to determine if a failure mode occurred. A web-based dashboard was used by the customer to monitor all their equipment and plan for service on equipment experiencing failures.

I designed the cloud infrastructure to support the back-end portion of the web-based dashboard in Amazon Web Services (AWS). IoT Core was used to handle all communications to and from the sensing systems using MQTT communication protocol. Every event the sensing system measured produced up to 250 packets of data that were sent to IoT Core. IoT Core sent the packets of data to Lambda which converted the binary packets of data into a numeric data type and stored it in a database in the Relational Database Service (RDS). CloudWatch was used to start a containerized application every 15 minutes that processed the data packets and stored optimized data in a Redis cluster in ElastiCache for the front-end portion of the application to access.

I developed the extract, transform, load (ETL) containerized process that processed all received data packets from the sensing system. The ETL process combined data packets, cleared events where all data packets were not received, and eliminated redundant data packets; stored copies of event data as files in Simple Storage Service (S3); created a data record in the database; fit a physical model to the data to derive performance parameters that predicted equipment failure; and updated an optimized data packet in Redis for the front end to quickly access. The data packet in Redis was a time series JSON formatted file of all data for a single IIoT unit with additional metadata such as statistics like average and standard deviation, specification and control limits, and anomaly detection. This allowed the front end to generate a time series control chart figure very rapidly for every unit a customer had deployed within their facility.

I optimized the physical model to extract the performance parameters from the time series data collected by the sensing system. The time series data was a high frequency measurement that could be fit using a piece-wise function with between 30 and 40 fitting parameters depending on the type of event that occurred. The type of event was determined by a supervised machine learning model which predicted the type of event, and the appropriate piece-wise function was fit to the time series data using differential evolution. Each fitting parameter was associated with a particular physical parameter of the piece of industrial equipment and could be used to monitor for several different failure modes as these performance parameters drifted outside of the specification limits.