|  |
| --- |
| UDITYA LAAD  [udityalaad123@gmail.com](mailto:udityalaad123@gmail.com) | (+1)519-729-9026 | Waterloo, Ontario, CA  <https://www.linkedin.com/in/uditya-laad-222680148> | <https://github.com/udityalaad> |

|  |
| --- |
| **PROJECTS**  **Package for 4 different Symbolic Execution Engines in WLANG –** (Explore SymEX)  **Jun 2022 – Aug 2022**   * Created Symbolic Execution Engines for – **Classical** Symbolic Execution, **Selective** Symbolic Execution (2 Versions), and **Dynamic** Symbolic Execution; using WLANG as the artifact. * Compared and analysed how each one fares against the other and where their specific utility may be applicable.   **Tech Stack:** Python, WLANG, TatSu, z3 Solver  **Repo:** <https://github.com/udityalaad/Explore_SymEx>  **GPS Based, Real-Time Accident Management System –** (Stay Safe) **Jul 2018 – Jul 2019**  An IoT project providing a collaborated solution for accident prevention, accident detection and rescue/aid – using a blend of Cross-platform applications, Microservices & Embedded Technology.   * Created a real time, multi-platform application - with unique dynamic functionalities for 3 sets of users. * Created Black Box Device to automatically detect accident of vehicles & integrated it with user application. * Created Web Services to implement algorithms & performed Data Analytics to provide better responses.   **Tech Stack:** Node.js, Cordova, Raspberry Pi, Python, Web Development, Firebase, Cloud Messaging, Cloud Functions, Android Application Development, Data Analytics  **Repo:** <https://github.com/udityalaad/GpsBased_RealTime_AccidentManagementSystem>  **A novel approach to System Design for Machine Learning –** (Travel Genius)  **Feb 2023 – Apr 2023**   * Created a short-term accommodation portal to demonstrate an innovative approach to System & DB Design, Query Optimization, Scalability, Data Mining and utilization of Machine Learning & LLMs in modern applications. * 3 major components include – Data Store, Client Application & leveraging Neural Networks to derive valuable insights.   **Tech Stack:** MySQL, Node.js, Express.js, React.js, Python, Tensorflow, Jest  **Repo:** <https://github.com/udityalaad/TravelGenius>  **Asynchronous Meal-Service Application with Model Optimization –** (FoodBell) **Jan 2022 – April 2022**  ‘FoodBell’ is an asynchronous solution for facilitating subscription-based (recurring) meal services – using a dedicated application, secure micro-service architecture and CI/CD approach.   * Created an optimal ‘vendor-consumer’ model & leveraged it to design the user application with targeted functionalities. * Architected a secure Micro-service model, with a Gateway facilitating 3 different services (consumer, vendor, auth). * Allowed real time communication between vendor & consumer platforms; used Cron jobs to facilitate automated changes.   **Tech Stack:** Spring Boot, MySql, Node.js, React, Java, Junit, Integration Test Framework (in react), Selenium.  **Repo:** <https://github.com/udityalaad/FoodBell>  **Optimizing Vertex-Cover Problem & Comparison with Other Methods  Mar 2022 – April 2022**   * Implemented and analysed 2 polynomial-time & 1 NP-complete algorithm for solving the minimum vertex-cover problem. * Optimized the existing encoding of CNF-SAT solver (NP-complete) to provide most optimal result at > 60% faster rate.   **Tech Stack:** C++, UNIX.  **Repo:** <https://github.com/udityalaad/OptimizingVertexCoverProblem>  **Sentiment Analysis using CNN  Jul 2022**   * A Convolutional Neural Network to perform Natural Language Processing (NLP) on varying input spaces, achieving > 80% accuracy in majority of cases; without the involvement of any word representation packages like word2Vec or Glove.   **Tech Stack:** Python, TensorFlow  **Repo:** <https://github.com/udityalaad/Sentiment_Analysis_Using_CNN>  **Automated Program Verification Engine for Imperative Languages –** (VeriCross)  **Jul 2022 – Aug 2022**   * Created a custom Program Verification engine to prove correctness using specifications & constructs like loop invariants. * The tool made deductive analysis possible with over 97% accuracy, as proved for over 10 test programs.   **Tech Stack:** Python, IMP, TatSu, z3 Solver  **Repo:** <https://github.com/udityalaad/VeriCross>  **KSOM based Recurrent Neural Network –** (ClustReduce) **Jun 2022**   * Created a custom Kohenen Self-organising Map from scratch, to cluster inputs & reduce dimensionality of input space. * Analysed the network's performance with increasing number of epochs for different configurations of the network.   **Tech Stack:** Python  **Repo:** <https://github.com/udityalaad/Clust_Reduce>  **Fuzzing Tool for Doom Video Game –** (Doom Fuzz) **May 2022 – Jun 2022**   * Created a fuzzing tool that tests and covers over 80% source code in Doom Video Game. * Used specialized techniques to create an effective entry point that ensures maximum reachability and coverage.   **Tech Stack:** C++, Linux, Sanitizers, CMake, Ninja  **Repo:** <https://github.com/udityalaad/Doom_Fuzz>  **Market Prediction using RNN  Jul 2022**   * Created Recurrent Neural Network for prediction of market stocks, achieving > 80% accuracy for over 5 sets of data.   **Tech Stack:** Python, TensorFlow  **Repo:** <https://github.com/udityalaad/Stock_Market_Prediction_Using_RNN>  **RBF Neural Network for Map Approximation  May 2022 – Jun 2022**   * A custom-configurable closed-form Radial Basis Neural Network to approximate mappings for interpolation purposes. * Analysed the network’s performance for different methods of center-approximation, with focus on significance of ‘spread’.   **Tech Stack:** Python  **Repo:** <https://github.com/udityalaad/RBF_NN_for_Map_Approximation>  **Feed-forward MLP classifier based on Gradient Descent  May 2022**   * Created a supervised feed-forward neural network for classification purposes, allowing over 50 custom options of configuration; with the ability to achieve over 85% accuracy for majority of viable configurations.   **Tech Stack:** Python  **Repo:** <https://github.com/udityalaad/FeedForward_MLP_Classifier>  **Street Specifications Generator & Optimal Path Finder – using Multiprogramming  Jan 2022 – Mar 2022**   * Designed street generator to create valid specifications in less than 10 unsuccessful attempts (further converted to graph). * Dynamically generated trails for specifications & facilitated communication between processes using multiprogramming.   **Tech Stack:** Python, C++, UNIX.  **Repo:** <https://github.com/udityalaad/StreetSpecificationsGenerator_and_OptimalPathFinder>  **Practical Implementer & Stepwise Analyzer for CCNS Techniques –** (Cyber Solve)  **Mar 2019**   * Developed an application to generate simplified results for complex techniques and algorithms used in ‘Computer Cryptography and Network Security (CCNS)’. * Also provided the option of detailed analysis, by allowing step-by-step tracing of each implementation.   **Tech Stack:** Java  **Repo:** <https://github.com/udityalaad/Cyber_Solve>  **Intelligent Analyzer & Suggestions Builder, cum E-Commerce Platform –** (The Shoe Rack)  **Apr 2018**   * Developed and implemented coherent algorithms to perform efficient cost and feature analysis, in order to generate dynamic comparison of products. * Created a Web App. - to display best-to-worst results, and also act as an E-Commerce portal (for shoes).   **Tech Stack:** Java, Web Development (JSP, HTML, JavaScript, CSS), SQL (Oracle Database), Apache Tomcat.  **Repo:** <https://github.com/udityalaad/The_Shoe_Rack>  **Applications for Simple Mini-Games**   **Feb 2018**   * Created applications for simple games like Ball Bounce (with smart Obstacles Generator), Maze Game (‘Self play’ + ‘Automated - with intelligent Path Decoder’), and Tic-Tac-Toe (‘v/s Player’ + ‘v/s Computer’).   **Tech Stack:** Android application development, C, C++  **Repo:** <https://github.com/udityalaad/Applications_for_Simple_MiniGames>  **Simple Reservation System for Airlines – (Go Fly)**   **Oct 2017**   * Created an easy-to-use application for flight reservation and payment, using Java & SQL (Oracle Database).   **Tech Stack:** Java, SQL  **Repo:** <https://github.com/udityalaad/Go_Fly> |