HARDIK RUPAREL

Irvine, CA 92612 | +1-949-994-2615 | hruparel@uci.edu | linkedin.com/in/hardik-ruparel | github.com/Hardik27

EDUCATION

University of California, Irvine — Irvine, CA
Master of Computer Science — December 2022
Anticipated Coursework: Advanced Programming and Problem Solving, Artificial Intelligence, Computer Security

Veermata Jijabai Technological Institute (VJTI) — Mumbai, India
Bachelor of Technology in Information Technology | GPA 3.67/4.0 — May 2019

TECHNICAL SKILLS

- Programming languages: Java, Python, PL/SQL, C++
- Web Technologies: HTML, CSS, JavaScript, JQuery, Bootstrap
- Domains & Miscellaneous Technologies: Machine Learning, Blockchain, Deep Learning, OOP, OOD, APIs, Git, Excel

PROFESSIONAL EXPERIENCE

UBS — Mumbai, India
Software Engineer — July 2019– July 2021

- Developed data loaders, processors and exporters using Java and PL/SQL for various Credit Risk calculations
- Utilized PySpark, Python and SQL to execute Credit Risk LGD calculations for the Basel III regulatory requirement
- Collaborated with a team of 5 to create a Question-Answering system on the Credit Risk regulatory data by implementing pre-trained BERT and Elasticsearch using Farm and Haystack Python libraries for improving the answer retrieval rate
- Accomplished team's goal of data migration to cloud using SQL to help facilitate the Phase-1 of the Credit Risk-wide Azure migration to reduce the IT operational cost
- Designed and executed "Eye Contact Detection" and "Balanced Smile Detection" components using Computer Vision and Python for a project- "Mock Interview Evaluator" to help underprivileged students prepare for job interviews
- Led a team of 4 to devise an online and a multiplayer version of "Meri Dukaan" to make the application more accessible and help underprivileged students improve financial literacy remotely

UBS — Mumbai, India
Software Engineer Intern — June 2018– July 2018

- Developed a mobile gaming application- "Meri Dukaan" utilizing JavaScript, CSS3 and HTML5 and in collaboration with an NGO- Pratham, to enhance financial literacy among underprivileged students
- Engineered the application according to the Reusability Design Principle by dividing the UI into templates to increase the modularity and to ensure easy maintenance of the codebase
- Expedited application delivery by ~2 months by following the Agile methodology, owing to which the application is currently live in India amongst 65,000+ active users

PROJECTS AND PUBLICATIONS

Hindi to English: Transformer-Based Neural Machine Translation ● Code ● Publication

- Trained the Transformer model for translating texts from Hindi to English using OpenNMT-tf library
- Performed various data cleaning and data pre-processing steps on the IIT Bombay English-Hindi Parallel Corpus, resulting in the training universe of 1.2 million parallel records
- Augmented the training universe by adding 3 million back-translated records from the monolingual WMT14 English News crawl dataset for expanding the model's vocabulary and knowledge-base
- Achieved a state-of-the-art BLEU score of 24.53 on the test data of 2400 parallel records

GeoSharding: A Machine Learning-Based Sharding Protocol ● Code ● Publication

- Designed a sharding protocol to optimize the speed and scalability of blockchain systems by dividing blockchain's network plane leveraging K-means algorithm on geographical location of miner nodes
- Formulated a novel leader election algorithm to elect a leader node in each shard to enhance the speed of blockchain
- Attained 200x faster shard creation speed as compared to traditional PoW-based sharding algorithm

Secure Voting for Democratic Elections: A Blockchain-Based Approach ● Publication

- Devised a novel voting architecture with advanced security measures such as multi-signature authentication and prevention of frequency analysis of hashes by leveraging benefits of blockchain and cryptography

Employee Attrition Rate Prediction ● Code

- Trained SGD Regressor to predict attrition rate on the training dataset comprising of 22 features and 7000 records
- Applied feature engineering techniques and achieved an accuracy of 81% on test data of 3000 records

Malicious URL Detection ● Code

- Performed various data pre-processing steps to extract 15 lexical features from the given URL
- Implemented Random Forest Classifier using scikit-learn and attained 90% accuracy on the test data of ~1 million records