




# NURRACHMAN LIU

Redwood Shores, California

☎ [415-988-0338](tel:415-988-0338) ✉ [rachliu@gmail.com](mailto:rachliu@gmail.com)  [rachliu](https://www.linkedin.com/in/rachliu)  [rliu-swe](https://github.com/rliu-swe)  [rliu-swe/code](https://codesandbox.io/s/rliu-swe/code)

## ALL PROJECTS


---

### ML/AI PROJECTS WITH PAPERS (GITHUB DIRECTORY )

---

**Quantized Neural Networks** [PDF](#)  | [Tensorflow](#), [Keras](#), [TF-QAT](#) 05-2022

- Implementation & analysis of quantized neural-networks. **Quantization of AlexNet, VGG16, ResNet50.**
- **Project Advisor:** Professor Adnan Darwiche

**Language Models, Embeddings, and HMMs** [PDF w/code](#)  | [Python](#), [NLTK](#), [SkLearn](#), [Gensim](#) 05-2022

- Impl. of language models, embeddings (sparse, dense, word2vec), and HMMs in context of Bayesian Networks.

**Fundamental Reinforcement Learning Algorithms** [PDF w/code](#)  | [Python](#), [OpenAI Gym](#) 05-2021

- Impl. of key RL algorithms: Q-Learning, Sarsa, Sarsa- $\lambda$ , UCB Exploration, Gradient Bandits, MC-Control Methods.

### ML/AI CODE-ONLY PROJECTS (GITHUB DIRECTORY )

---

**Bayesian Network Toolkit for Inference, Learning, and More** [Github](#)  | [Java](#) 03-2022

- Implementation of Bayesian Network operations such as Factor Multiplication, Marginalization, etc.
- Implementation of Bayesian Network algorithms such as Variable Elimination (VE), Factor Elimination (FE), Join-Trees, Elimination-Trees, Interaction-Graphs
- To-do: Most Probable Explanation (MPE) (i.e., Viterbi with a specific VE and when BN is HMM); Maximum a-posteriori Hypothesis; Belief-Propagation (Exact and Approximate); Bayesian-Network Model Learning

**Data Mining Toolkit for Frequent, Sequential, Tree and Graph Pattern Mining** [Github](#)  | [Java](#) 04-2022

- Frequent Itemset Pattern Mining: Implementation of Apriori algorithm, Candidate-Generation for Itemsets, CLOSET (Mining of frequent closed patterns), DIC (Dynamic Item Counting), FP-Growth, Max-Miner
- Frequent Sequential Pattern Mining: Implementation of GSP, Candidate-Generation for Sequential patterns, PrefixSpan
- To-do: Advanced Clustering Algorithms: Medoids based: PAM; Medoids-Sampling based: CLARA, CLARANS; Hierarchical Clustering: BIRCH; Density-based clustering: DBSCAN, STING, OPTICS
- To-do: Association-Rule Based Classifiers: CAR & M1 (done in Python in another repo)
- To-do: Frequent Tree Mining: TreeMiner
- To-do: Frequent Graph Mining: FSG, gSpan, FFSM