Dang Le Dang Khoa

(+65) 8398 3951
dangledangkhoa@gmail.com
github.com/dangkhoadl
linkedin.com/in/dangledangkhoa

Language Proficiency: **Python, C/C++, Bash, SQL, Scala-Spark**

Professional Experience (selected)

Research Engineer **Institute for Infocomm Research - A*STAR** **February 2020 - Present**

Speech-to-text engines:

- Bahasa Malay Automatic-Speech-Recognition (ASR) - Core ASR Engine project
  - Train and maintain deep learning Speech-to-Text and language model, improve 10% in terms of word-error-rate (WER) compared to the former production model.
  - Clean and maintain Malay language data, including audio, text, dictionary formats.
  - Explore industrial applications of state-of-the art transformer-based, end-to-end Speech-to-Text engines.
- Air Traffic Control (ATC) ASR
  - Pre-process and prepare raw audio/text data for training.
  - Apply speech enhancement model to denoise audio data for other subsequence tasks:
    1. Enhance Voice activity detection (VAD).
    2. Enhance raw Audio quality for labeling task.
    3. Enhance output Audio quality to improve client listening experience.
- Legal-domain ASR
  - Pre-process and prepare raw audio, text data for model training (remove bad quality audios, parse legal-specific documents to cleaned text data).
  - Benchmark and evaluate final production models.
  - Visualize model efficiency and data statistics to end-user.
- *Toolkits*: **kaldi, Espnet, flashlight-wav2letter, Docker**

Natural Language Processing - Audio Analytics and Classifications:

- Speech enhancement deep learning model
  - Implement bi-LSTM Densenet and U-net models.
  - Train with LibriSpeech, VoxCeleb, Aishell, MUSAN datasets.
  - Apply the model to denoise Air Traffic Control Audio data, improve noised audio quality by 25% in terms of snr, pesq, and stoi.
- Speech classification applications
  - Train vector embedding and classifier to identify Covid-19 patients from coughing sound, resulting auc = 0.85.
  - Train model to classify deep-faked audios from authentic audios.
- *Toolkits*: **PyTorch, XGBoost, Scipy stacks**

Others tasks:

- Build and maintain training pipeline and docker templates for the team.
- Train new staff and exchange experience on ASR engine, data wrangling & collecting, and PyTorch framework topics.

Researcher **Satellite Research Centre** **August 2016 - August 2018**

• Implemented a design of Star Tracking Algorithms onto a Programmable System-on-a-chip system.
• Optimized the pattern recognition algorithm runtime by implementing the connected component labeling algorithm on parallel processors. Runtime is improved 64% on average compared to traditional processor approach.
• Optimized and designed the pattern searching algorithm by applying k-ary tree data structure. Time complexity and runtime improved 31%, but space complexity and memory increased 22%.
Source code: git.io/vpucY
Publication: https://doi.org/10.32657/10220/48371

Education

Master of Engineering **Nanyang Technological University** **August 2015 - August 2018**

• School of Electrical and Electronics Engineering - Research.
• Thesis title: Implementation of An Autonomous Star Recognition Algorithm using Hardware-Software Co-Processing Approach.

Bachelor of Engineering **Vietnam National University HCMC** **August 2010 - April 2015**

• Ho Chi Minh City University of Technology - Electrical and Electronics Engineering - Second Upper Honour.
• Major in Automation and Control engineering, minor in Robotics and Embedded System Design.
• Thesis title: Applying of Fuzzy Logic Algorithm on Legged Locomotion Robot.

Open-Source Projects (selected)

WER-in-CPP

- Develop an open-source API to calculate Word-Error-Rate for ASR project based on Minimum-Edit-Distance problem.
- Develop new features compared to existing kaldi code: Provide WER-per-utterance utilizing upon dataset statistical analysis.

Technologies: **C++**
Source code: https://github.com/dangkhoadl/WER-in-cpp

Problem Solving

- Solve Computer Science and Competitive Programming Problems.

Source code: https://github.com/dangkhoadl/my-CS-Notebook

- Solve Machine Learning Problems.

Source code: https://github.com/dangkhoadl/my-Machine-Learning

Stock Price Predictor

- Predicted Stock Price data by Linear Regression and ARIMA modeling approaches.
- Optimized Model's accuracy by 20% by performing feature engineerings and hyper-parameters optimizations.
- Evaluated and Backtested models based on Live Stock Price Simulator on MetaTrader4.

Financial Knowledge:
Coursera - Introduction to Financial Markets
Coursera - Portfolio and Risk Management
Technologies: **pandas, numpy, matplotlib, sklearn**
Source code: git.io/vpuCA

Dota 2 Hero Recommender System

- Recommended game characters based on historical user data collected by Open Dota API.

Technologies: **flask, pandas, numpy**
Source code: git.io/fhV1q

Certificates and Related Courseworks

- Coursera - Data Structures and Algorithms Specialization, Machine Learning Specialization, Mathematics for Machine Learning, Introduction to Deep Learning, Big Data Essentials: HDFS, MapReduce, and Spark RDD
- Udacity - Machine Learning Engineer Nanodegree
- MIT-6.041-Probabilistic Systems Analysis and Applied Probability
- UC Berkeley - CS162 - Operating Systems and Systems Programming
- UIUC - CS425 - Distributed Systems

Competition Achievements

Google Code Jam 2018

- An annual contest for algorithmic problem solving, hosted by Google
- Round 2 qualifier - Top 10% Candidates

ConferencingSpeech 2021 Challenge

- A Challenge for Far-field Multi-Channel Speech Enhancement for Video Conferencing, hosted by Interspeech and Tencent Ethereal Audio Lab
- Member of I2R-ALI team - Top 10 ranking team