

**Name:** Mr David Goh Zhe Kai    **Gender:** M    **HP:** +65-92357049  
**E-mail:** david.gohzhekai@gmail.com

## Experiences

- Sep 2020 – Jan 2021    *Gracet.AI*, Intern – Full Stack Software Developer
- Worked in a multi-cultural team to deliver a HR management product
  - Proposed and implemented the testing flow of the GraphQL resolvers through Jest and Node.JS
  - Implemented GraphQL resolver for attendance on Express to communicate with the MongoDB server using Mongoose.
  - Crafted input verification for resolvers using Yup
  - Built the full stack development of the email verification using React in the frontend, Apollo to communicate with the backend and GraphQL in the backend.
  - Proposed RBAC framework for authorisation
  - Designed the error handling flow for the back end and communicated the error with the front end
  - Developed the Contact Us page and integrated with the backend
- Oct 2020 – Dec 2020    *NUS High School, AppVenture, nush.link*, Lead developer  
*nush.link* (<https://github.com/appventure-nush/nush-link>)  
*nush.link* is a URL shortner website for students and teachers from NUS High School to use to shorten their URLs.
- Oversaw and led the completion of the project
  - Built a RESTful backend using TypeScript and Express on Node JS and MySQL server as the database to communicate with the front end
- Dec 2020    *GovTech, GeekOut*, Participant  
*Chope (Hackathon)*  
*Chope* is a device which allows those with mobility difficulties to have easy access to critical services.
- Led a group of six to make a hardware and website POC
  - Assigned members roles and responsibility
  - Facilitated the Sprint meetings during discussion
  - Pitched the project in front of senior developers
  - Worked on the communication between devices using the internet using Arduino, ESP8266 and MQTT
- Aug 2020    *NUSH Hacks '20*, Backend Developer  
*nush.link (Hackathon)*  
*nush.link* is a URL shortner website for students and teachers from NUS High School to use to shorten their URLs.
- Oversaw the development and ideation of the project to develop a POC
  - Created a MySQL database and connect it to the backend
  - Used Express.js on Node.js to develop the backend and the creation of API
- Jul 2020 – Nov 2020    *NUS High School*, Mentor Representative  
*Sharing with Year 1 and Year 2 juniors*
- Rallied the class to organise a sharing session for the lower year students
  - Led the discussion and managed a team of twelve to share with the lower years
  - Discussed with multiple stakeholders to organise the sharing

- Jan 2019 – *DSO National Laboratories, Intern - Research*  
Jan 2020 *Information Extraction from Air Traffic Control Communications*  
A research project which aims to use Machine Learning in Speech to Text to extract out key information from Air Traffic Control Communications
- Oversaw and led the completion of the project
  - Applied machine learning and neural network to improve the language model and entity extraction in speech to text
  - Utilised FRAGE, a Long Short-Term Memory (LSTM) neural network, as the language model
  - Wrote code using Python to parse lattices
  - Updated language scores in lattices using a LSTM language model instead of N-gram language model to achieve better accuracy
  - Implemented beam search for lattices
- Nov 2019 *NUS High School, AppVenture, Introduction to Python Workshop, Facilitator – Curriculum Planner*
- Assigned by school's Computer Science interest group to facilitate a 3 hours workshop for 20 students ranging from Year 4 to Year 6
  - Developed the slides for the workshop
  - Set the questions on Coursemology, a platform for educators, for participants to practice during the workshop
  - Equipped the participants with the foundation knowledge to explore more about Python and programming
  - Facilitated the session during the workshop
- Jan 2018 – *DSO National Laboratories, Intern - Research*  
Jan 2019 *Decompilation of Assembly Language using Machine Learning*  
A research project which used machine learning to reverse engineer assembly language for better analysis of the program
- Oversaw the arrangement of meeting and direction for the project.
  - Used IDA Pro, a disassembler, to view and process assembly code
  - Wrote a Python program that used CParser, a Python module to parse C, to standardise C code.
  - Decompiled and evaluated on Open Source project (OpenSSL and Redis)
  - Managed to achieve high scores for synthetic dataset to show that neural network can decompile simple C functions.
  - Model could decompile well on short C functions from real world project.
- Jun 2017 – *DSO National Laboratories, Intern - Research*  
Jan 2018 *Exploring Decompilation of Assembly Language using Pointer-Generator Networks*  
A research project which used machine learning to reverse engineer assembly language for better analysis of the program
- Installed Pointer-Generator Networks, a Natural Language Processing neural network, on a Linux server
  - Wrote Java code to synthesise simple C functions
  - Used GCC to compile C functions to produce assembly code
  - Paired the C functions with assembly code and fed into neural network
  - Evaluated the neural network output using the ROGUE metrics and produced scores over 95 for all ROGUE metrics

## **Skills**

### *Cyber-Security*

- Joined team “CS\_Gang” which was 6<sup>th</sup> in Singapore at CTFTIME in July 2020 (2020)
- Completed cyber-security courses offered by Immersive Labs (2019)
- Participated in Young Defence Scientists Programme (YDSP) World of Science – Computer Security (2018)

### *Machine Learning and Artificial Intelligence*

- Wrote a Jupyter Notebook which uses different machine learning techniques to determine the nature of a program based on the imported libraries in Windows (2020)
- Produced a 4D prediction program using neural network and genetic algorithm for school project (2020)
- Completed multiple DataCamp courses on machine learning and AI (2019-2020)
- Participated in Young Defence Scientists Programme (YDSP) World of Science – Artificial Intelligence (2018)

### *Presentation*

- Student Helper for Computer Science Booth for NUS High School Open House (2019)
- Peer tutoring at Pei Tong Primary School (2017)

### *Competitive Programming*

- Completed a school module on Algorithms and Data Structures (2019)

### *Computer Application*

- Created a computer game using Java OOP and GUI for school project (2017)

### *Full Stack Development*

- Built a website for Hackathon to allow students to shorten their URL that used React.js as the front end and Express.js as the backend and MySQL as the database. (2020)
- Built a website for police database on criminals and offences for school project that used HTML, CSS, JS as well as backend PHP and MySQL (2019)

### *Android App Development*

- Made a To-Do Android app for school module project (2018)

### *Microcontroller*

- Produced a GUI in Python to control LED lights from a Raspberry Pi (2016)

## Achievements

- 2020
  - DSO Star Award for achieving A- in NUS CS2100 module
  - 11<sup>th</sup> in STACK the Flags 2020 (CTF)
  - 17<sup>th</sup> in Cyber Defenders Discovery Camp (CTF)
  - 9<sup>th</sup> in WhiteHacks2020 (CTF)
  - Participation in NUSH Hacks'20 (Hackathon)
  - Participation in NUS Department of Statistic & Applied Probability 2020 Statistics Competition
  - NUS High School Leadership Commendation Award for Mentor Representative
- 2019
  - Singapore Science and Engineering Fair Finalist
  - DSO Star Award for achieving A in NUS CS1010S module
  - Participation in Cyberthon (CTF)
  - Bronze in Cyber Defenders Discovery Camp (CTF)
  - Participation in Australia Informatics Olympiad
  - NUS High School CCA Commendation Award for Soccer CCA
- 2018
  - Merit in Science Mentorship Program
  - Presented at Raffles Institution Research Education Congress 2018 at the Interactive Session Project Presentation
- 2017
  - NUS High School CCA Commendation Award for Soccer CCA
  - NUS High School Service Commendation Award
  - Edusave Scholarships for Integrated Programme Schools (ESIP)
- 2016
  - Clinched best algorithm award in building an Arduino robot to navigate through the maze
- 2015
  - Achieved a Distinction in Beaver Computing Challenge
  - Edusave Good Progress Award

## Educational Qualification

- 2015-2020
  - *National University of Singapore High School of Maths and Science Diploma* with Major in Chemistry and Honours in Mathematics and Computer Science
    - Python: Basics, OOP, GUI, Raspberry Pi
    - Java: Basics, OOP, GUI, Android
    - Web development: HTML, CSS, JS, PHP, MySQL
    - Algorithms and Data Structures
    - AI: Genetic algorithms, Machine learning, Neural Network, Fuzzy Logic
    - Computer Networking
- 2018-2020
  - *National University of Singapore*  
Completion of the modules
    - CS2106 Introduction to Operating Systems (2020)
    - CS2100 Computer Organisation (2019)
    - CS1010S Programming Methodology (2018)

## **Referees**

### **Dr Chieu Hai Leong**

Distinguished Member of Technical Staff

Information Division

DSO National Laboratories

(Mentor for *Information Extraction from Air Traffic Control Communications*, *Decompilation of Assembly Language using Machine Learning* and *Exploring Decompilation of Assembly Language using Pointer-Generator Networks*)

### **Dr Khoo Wei Ming**

Senior Member of Technical Staff

Information Division

DSO National Laboratories

(Mentor for *Decompilation of Assembly Language using Machine Learning* and *Exploring Decompilation of Assembly Language using Pointer-Generator Networks*)

### **Mr David Lim**

Chief Executive Officer

Gracet.ai

## **Samples of project**

### Information Extraction from Air Traffic Control Communications

Air Traffic Control (ATC) communication refers to the relay of crucial information between pilot and ground-based ATC personnel. Conveying this information is paramount in avoiding collisions with other planes, allowing a swift, efficient and safe transit.

Speech recognition in ATC will help mitigate the issue of increasing air traffic by allowing ATC personnel to focus on more critical situations. As air traffic becomes heavier, this increases the workload on ATC personnel and hinders them from focusing on more critical situations.

Speech-to-Text (STT) is a fundamental step in achieving speech recognition in ATC communications. However, this is a difficult problem due to the amalgamation of different issues such as high speech rate and noisy communication channels.

In this project, we propose using a State of The Art (SOTA) language model with different heuristics to reduce word error rate in lattice searching. In addition, we used a SOTA entity extraction tool along with Hidden Markov Model in identifying callsigns to improve STT accuracy.

Our results reveal that using a different language model provides a better transcription as compared to language model provided by STT toolkit. In addition, we have shown using Hidden Markov Models can improve Conditional Random Field sequence models in correcting extracted callsigns.

To advance the current work, we purpose using different language models and sophisticated Named Entity Recogniser in tandem with more training data to improve the accuracy of STT in ATC.

### Decompilation of Assembly Programs using Machine Learning

In this project, we present a novel technique of using Pointer-Generator Network (PGN), a modified Sequence-to-Sequence attentional model, to decompile code into its equivalent C program. The viability and efficacy of PGN for decompilation were examined. Due to the complex and inefficient nature of traditional decompilation techniques, we present PGN as a possible alternative to traditional decompilation. This technique has the potential to eliminate problems that current compilers face such as compiler optimisations or the production of source code that is difficult for humans to read. As the PGN is trained on human-written code, the generated code from the PGN would hence be similar to human-written code. This allows the researchers to understand the logic of the program better. In this project, we trained the PGN on simple code as well as real-world code. To simplify the dataset for the neural network, various pre-processing techniques were examined. We concluded that PGN was able to decompile simple cases of assembly which it had not encountered before. When tested on real-world code, preliminary experimentation has shown that PGN was not able to decompile well. Hence, more work is needed to determine the full extent of decompilation using PGN.

### Exploring Decompilation of Assembly Language using Pointer-Generator Networks

Currently traditional decompilation is done through rule-based procedures. Due to the exceptions that can arise in code, traditional decompilers must be programmed to catch these exceptions. As a result, decompilation is very complicated and the generated code is prone to error. In this report, we used a novel approach of using neural networks to decompile. The neural network was trained in assembly language to generate C code. The generated C code was evaluated using ROUGE to determine the accuracy of the neural network. Our results have shown that the generated C code shows a high level of accuracy on simple C programs containing assignment and basic operators. Hence, this experiment concluded that neural networks can do decompilation. However, in the future, more research needs to be done to determine how feasible neural networks are in decompilation on more complex C programs.