

ZIHAN ZHANG

✉ zhangzihan_96@outlook.com · ☎ (+61) 0401-115-299 · 🌐 Github

🎓 EDUCATION

University of Technology Sydney, Sydney, Australia 2021.3 – Present

PhD student in Engineering and IT

- social data analysis, natural language processing, machine learning

University of Melbourne, Melbourne, Australia 2018.7 – 2020.12

Master student in Software Engineering, expected December 2020

- GPA: 83/100 (First Class Honours)
- Awards: Dean's Honour List 2019

China Pharmaceutical University, Nanjing, China 2014.9 – 2018.6

BMgmt. in Information System and Information Management

- GPA: 82/100 (Ranking: 25/143)
- Awards: Second-Class Scholarship (2016-2017) Excellent Volunteer (2015-2016)
- Activities: President of the Faculty of Science Student Union (2016-2017)

👥 WORKING EXPERIENCE

TPG Telecom Sydney, Australia 2021.3 – Present

Research Analyst Fulltime

As a part of the Business Intelligence group, I transform numerous data into actionable insights.

- **NPS topic modelling** - extract keywords from customers' feedback and study why the customers are satisfying/unsatisfying.
- **Market offer text extraction** - extract and analyse competitors' offers, transform from raw images to structured data that the market team could use.
- **Business Insights analysis** - model and analysis customers churn and upgrade
- AWS ML pipeline

RESORTer Melbourne, Australia 2019.12 – 2020.3

Front-end Developer Intern

I was responsible for refactoring and developing the Lesson Section in the Web application.

- Refactored the Lesson Section using React + Hooks + Material-UI. Used Grid layout and Card component to render different kinds of lessons and simplified the rendering logic, which was a serious issue when previously using the Tabs system.
- Managed the global state using Redux and created default lessons for the users based on the form that they filled. I also cooperated with my team using Middleware to catch and handle certain actions to make sure the generated lessons are always consistent with the global state, thereby improving the user experience.
- Utilized CSS Module to avoid class names collisions and global style pollution. Used lazy load to dynamically import required components, thereby improving the performance.

📁 RELEVANT PROJECTS

Short Text Location Identification

Some supervised Machine Learning methods were built, compared and analysed with the aim of automatically identifying the location from which tweet messages was sent.


- Given the raw dataset, the Term Frequency - Inverse Document Frequency (TF-IDF) algorithm was used to weight features and built a word corpus.

- Bernoulli Naïve Bayes (BNB), Logistic Regression (LR) and ensemble methods Stacking were used to train models. These models were compared and analysed using learning curves diagrams and confusion matrix.
- The model achieved a high accuracy that ranked in the top 30 out of 120 in the final Kaggle tournament.

Guttman Chart Analysis System

 [Github](#)

A Guttman chart based students assessment analysis system. It can be used to help educators find students' Zones of Personal Development(ZPD) and adjust future teaching plannings.

- The project provided support for the  research in the Assessment Research Centre, Melbourne Graduate School of Education.
- To detect irregular patterns in the data, an algorithm was developed to calculate the local similarity of each item and detect any irregular ones.
- I was responsible for developing the frontend pages and integrating it with the backend developers. The project used Python as the backend programming language, adopted the Client/Server model, and used RESTful API for HTTP communication.

Algorithm in Action

Demo: <https://algorithm-in-action.herokuapp.com/>

An algorithm visualization Web application provided for the first year Computer Science students.

I was responsible for implementing the pseudocode and algorithm animation.

- Using JavaScript function closures, all the visualization API functions and corresponding variables can be stored in an array and executed later, so it solved the problem of using ES6 Generators that functions executions cannot be reversed. Thereby, the animation can step backward as well.
- To map the algorithm pseudocode with the actual code, I parsed and added a bookmark in each line of the pseudocode, and inserted the bookmarks at the corresponding position in the actual code, so the pseudocode and animation can be synchronized.
- Implemented a customized hook useInterval so that the auto-play function can read fresh states between each render. This hook can also detect the speed changes and reset the setInterval function, thereby adjusting the playback speed is achievable.
- Based on the visualization APIs provided by Tracer.js, I implemented some common components and functions and expanded the library as well.

Distributed Shared Whiteboard

 [Github](#)

A shared whiteboard desktop application that allows multiple users to draw shapes and chat at the same time.

I was responsible for developing the client and server GUI.

- The project used Java 8 as the backend language, JavaFX as the frontend framework, and used a three-tier Client/Server architecture. It separated the client – whiteboard server – data server.
- Java RMI was used as the communication method between the whiteboard server and data server, the request sends from the client were remotely called in the whiteboard server as well. To synchronize each client, MQTT was used to provide a subscribe/publish protocol. The whiteboard server was used as an intermediate agent to accept messages from each client and publish the messages to all other subscribers.

SKILLS

- Familiar with frontend knowledge:
 - Familiar with React and relevant techniques: Hooks, Redux and React-Router
 - Familiar with UI frameworks including Antd and Material-UI, as well as CSS preprocessors Sass
 - Understand common optimization methods like lazy load, tree-shaking, etc.
- Familiar with common data structures and algorithms
- Familiar with computer network knowledge
 - Familiar with HTTP, TCP/IP protocols
 - Understand common cryptography methods
 - Understand common web security attacks
- Programming Language: JavaScript > Java == Python > Haskell
- Other:

- Familiar with Git
- Understand CI/CD

i REFERENCE

Reference available on request.