

# Tate Cheng

# Software Developer

✉ [tl2cheng@edu.uwaterloo.ca](mailto:tl2cheng@edu.uwaterloo.ca) 🌐 [tate1010.github.io](https://tate1010.github.io) ☎ 647-402-6181 in [tate-cheng](#) 🌐 [tate1010](#)

## + skills

PROGRAMMING	Git, Python, C++, Java, Bash, SQL
TOOLS	Pandas, Numpy, XGBoost, Keras, TensorFlow, PyTorch, SK-Learn
SPOKEN LANGUAGES	English, Cantonese, Mandarin
OTHERS	Agile, OOP, DevOps

## + employment

**Data Analyst Student Intern** 88 Queen Quay West, Toronto  
**Royal Bank of Canada**  
May 2018 to Current

- Developed an analytic test automation suite, which finds and checks if events and their tags are being fired correctly into the database, for the Royal Bank of Canada Mobile App using **Python**, **Android Studio**, **Google Analytics**, and **BigQuery**
- Utilized the analytic test automation suite to find analytical errors for the upcoming 2.0 release of the mobile app
- Assisted on repairing the mobile app fingerprint authentication issue currently on the production line by pulling specific users' data which matches certain criteria using **Google's BigQuery** and Python and performed case analysis on these users' cases to determine the cause of the issue

**Machine Learning Research Assistant** Central, HongKong  
**Epiphany Asset Management (HK) Limited**  
May 2017 to July 2017

- Compared different machine learning algorithm models' performances and determined which model is more suitable for usage by building and developing deep-learning neural networks such as **LSTM**, **gradient boosting regression**, and **hybrid ARIMA** using HSI's data
- Reconstructed daily returns of HSI from monthly data and studied the properties of the reconstructed time series with the original one by utilizing compressed sensing and machine learning algorithms
- Compared the correlation between different hyperparameters on a model by performing hyperparameter optimization using grid search on different neural network models

**Computer Science Tutor** Waterloo, ON  
Sept. 2015 to Sept. 2017

- Worked with first year students at the University of Waterloo who were enrolled in CS 135, CS 136, CS 115, and CS 116
- Assisted students preparing for midterm exams by creating sample questions along with step-by-step thought processes for each solution and reviewing past solutions from assignments

**Computer Technician** Markham, ON  
**Mac&PC Repair Depot**  
Sept. 2015 to Jan. 2016

- Used analytical skills and knowledge of computers to determine issues and find appropriate technical solutions
- Repaired liquid-damaged, broken screen, and bricked MacBooks
- Installed, upgraded, and troubleshoot issues for software and hardware at user request

## + awards

University of Waterloo President's Scholarship	June 2016
University of Waterloo Faculty of Math Euclid Contest Scholarship	June 2016
Canadian Senior Mathematics Contest Distinction (top 25%)	Jan. 2016

## + education

**3A - University of Waterloo**  
Candidate For Bachelor of Computer Science 2020

## + hackathons

### RUHacks2018

- Best Green(Money) Hack**
- Built Cocoa, a proactive budget financial app that notifies you prior to making purchases at checkout
- Worked on the backend server with **Python**, **Flask**, and **Google Places API**

### ConuHack 2018

- First Place API Challenger Winner and Hackathon Overall Third Place Winner**
- Built Awesome sport, an automatically soccer game highlighter using **Data Analytic** and **Machine Learning**.
- Worked on extracting data using the **Astucemedia API** in the backend server

### YHack 2017

- Built Emotionji, a conversation analysis that is able to show the emotion between the two people at the time of speech
- Worked on the backend server with **Python** and **Google cloud speech**, **IBM Watson**, and **Iconic Machine Learning API**

## + projects

### Curtain Automation

- Attempted to build an automatic curtain opener that operates based on command or time
- Built with **Arduino uno Microcontroller**, **A3967 microstepping driver**, and **3D-printed part**

### QuadRis

- Built the retro game Tetris featuring custom-made rules and level using C++ and OOP
- Implemented a prefix trie and completed an interpreter that allows the user to shorten or repeat sequence of commands with ease

### Party Web Service

- Used **React**, **Material UI**, **Redux**, and **Node.Js** and built a web-based party game engine
- Focused on developing user experience
- Designed a unique feature that promotes physical interaction between players