

# KUANG-YU LI

IT Master Student focus on Cloud Computing

Allmandring 20D, Stuttgart, Germany 70569 | +49 152 07439908

kuangyu.li@outlook.com | LinkedIn/kuang-yu-li-lumiere | GitHub/kuangyu0801

## EDUCATION

<b>Universität Stuttgart</b> – Stuttgart, Germany	Oct. 2019 - present
<i>M.S. in Information Technology, German Grading: 1.8 (Gut)</i>	
<b>National Chiao Tung University (NCTU)</b> – Hsinchu, Taiwan	Sept. 2009 - Oct. 2015
<i>B.S. in Electrical Engineering and Computer Science, GPA: 3.9/4.3 M.S. in Electronics Engineering and Electronics, GPA: 4.27/4.3</i>	

## EXPERIENCE

<b>IBM</b> – Böblingen, Germany	Apr. 2021 - present
<i>Intern, IBM Systems</i>	
<ul style="list-style-type: none"><li>Developing an integrated development environment with remote development and debugging capabilities, static code analysis and unit test integration.</li></ul>	
<b>MediaTek, Inc.</b> – Hsinchu, Taiwan	Dec. 2015 - Aug. 2019
<i>Firmware Engineer, Communication System Design</i>	
<ul style="list-style-type: none"><li>Developed 4G and 5G digital signal processor (DSP) firmware in Android smartphone for mobile communication in physical (PHY) layer</li><li>Wrote 6K code lines, reviewed and maintained 20K code lines in C/C++ on Red Hat Enterprise Linux</li><li>Designed firmware architecture, implemented algorithm, developed and published tests with Python and Perl scripts for performance verification</li><li>Participated in 3 large-scale projects (over 200K code lines and 1000 engineers) and collaborated with cross-team SW and HW developers in 9 countries</li><li>Involved in all stages of SDLC for over 10 MediaTek's smartphone products including world's fastest 5G Helio M70 with download speed 4.7Gbps in 2019</li><li>Handled technical change requests with troubleshooting, log examination, issue analysis, solution implementation and patch releasing for global including Samsung, LG and Huawei, etc.</li><li>Received 7 times MediaTek vAward in recognition of top 10% performance of the month</li></ul>	

## TECHNICAL STRENGTHS

<b>Programming</b>	Java, Python, C/C++, JSON, HTML/CSS, SQL, Perl, MATLAB, Assembly
<b>Tool</b>	Git, GitHub, Perforce, IntelliJ, PyCharm, Android Studio, VirtualBox, Docker
<b>Platform &amp; Protocol</b>	Google Firebase, JavaServer Pages (JSP), OpenFlow, TCP/IP, HTTP, REST

## PROJECTS

<b>Content-Based Router for Publish-Subscribe Service</b>	Java, REST API, HTTP	SDN Lab
<ul style="list-style-type: none"><li>Developed a routing controller in a Java module, which provides content-based routing for pub/sub service to minimize network traffic and reduce application filtering effort. The routing algorithm is based on sorting and merging interval of encoded IP-address</li><li>Developed a "Subscriber" Java application which can subscribe to publishing service via REST API and receive UDP from publisher.</li><li>Developed a one-to-many publishing service in Floodlight controller, which can receive subscription via HTTP request (GET, POST, and DELETE) and perform content-based routing in a OpenFlow network.</li></ul>		
<b>Dynamic Routing for Software-Defined Network</b>	Java, Dijkstra's algorithm	SDN Lab
<ul style="list-style-type: none"><li>Developed a Java module in Floodlight controller, which provides 2 dynamic routing modes in OpenFlow network.</li><li>Reactive mode routes with shortest path. Adaptive mode routes TCP flow with load balancing by querying network traffic statistics dynamically and matching IP addresses and TCP ports. The implementation is based on Dijkstra's algorithm.</li><li>Verified application with Iperf in MiniNet on Linux and achieves 6x bandwidth increase (582kbs vs 3478kbs) in adaptive mode</li></ul>		
<b>Java Application for Wireless Ad-hoc Network</b>	Java, UDP, Raspberry Pi	Mobile Computing
<ul style="list-style-type: none"><li>Developed 4 Java server and client applications which implement 2 protocols: Flooding and Dynamic Source Routing (DSR). Flooding achieves high robustness with UDP messages broadcast. DSR achieves reduced data transfer overhead with route discovery in control messages. Applications use DatagramSocket classes from java.net package for UDP transmission.</li><li>Verified applications on Raspberry Pi with real mesh 802.11 WiFi network.</li></ul>		
<b>Android App for City Temperature with Google Firebase</b>	Java, Android	Mobile Computing
<ul style="list-style-type: none"><li>Developed an Android application, which can update, subscribe, and calculate daily average of designated city temperature</li><li>Implemented functions for accessing and querying data in JSON in a shared Realtime NoSQL database with Google Firebase API</li></ul>		
<b>ATP Tennis Player Network Analysis</b>	Python, Graph, NetworkX	Complex Network System

- Developed Python programs to generate complex network and derive structural insights such as Page Rank, Connectivity, Clustering, etc.
- Implemented algorithms with NetworkX package and built an undirected graph by processing real tennis match statics in csv format.
- Discovered, visualized, rendered and exported network topology with open-source software Gephi

### **Forest Cover Type Prediction – Python**

Machine Learning

- Implemented *Decision Tree* and *Support Vector Machine* with *Scikits-Learn* package, evaluated and discussed the performance on forest type classification problem