

# JIANFENG CHEN

<http://www4.ncsu.edu/~jchen37>

919-457-2034 ◊ [jchen37@ncsu.edu](mailto:jchen37@ncsu.edu) ◊ [github: ginfung](https://github.com/ginfung) ◊ [linkedin: janfengcs](https://www.linkedin.com/in/janfengcs)

## EDUCATION

---

### PhD in Computer Science

Aug 2014 - Dec 2018 (expected)

North Carolina State University, GPA: 4.0/4.0

Coursework: Data-to-Knowledge | DevOps | Advanced AI | Algorithm Analysis | Data Mining | Automated SE

### BS in Computer Science

Sep 2010 - May 2014

Shandong University, China, GPA: 91.1/100

Coursework: Data Structure | OS | Networking | Database System | Numerical Analysis | Image Processing

## SKILLS AND INTERESTS

---

Proficient in **language**: Python, Java; familiar with JavaScript, C++, Matlab and SQL;

Proficient in **data analysis** tools: Scikit-learn, SciPy, Pandas, jMetal, Gephi;

Familiar with **DevOps** tools: Jenkins, Ansible, Travis-CI, AWS Elasticsearch, S3, Docker, Redis;

Interested in machine learning/data related positions as well as backend development and (automated) testing.

## WORKING EXPERIENCE

---

### (Intern) Brahms Model Verification with Java Pathfinder platform

May 2016 - Aug 2016

*Google Summer of Code program 2016*

- Accepted by Google GSoc2016 program among 18,981 applicants (**accept rate: 6%**).
- Hierarchical clustering a dataset(flight log) with more than 20M entries top-down and bottom-up. Create a PCA-like dimension reduction algorithm and speed it up by spark. Compared my own algorithm with PCA.  
*Skills: Spark, Multi-objectives Optimization, Dimensionality Reduction, HAC*

### (Research Assistant) Sampling vs. Searching in Search-based SE

Dec 2014 - Present

*North Caroline State University*

- Created a sampling technique to solve the software product line problem. Deployed the algorithm into platform LSF by MPI; implemented a job schedule engine. Reduced the experiment time from 2 months CPU hrs into 11.5 hrs.
- Modeled the Linux Kernel modules in CNF sets. Created the decision tree surrogate model. By combining SAT solvers, found a way to configure large software systems 2000x faster. Published the results in TSE.  
*Skills: MPI, SBSE, Software Architecture Analysis, Z3 Solver*

## SELECTED PROJECTS

---

### LACE Data Privatization Tools and its Application

Aug 2016 - Nov 2016

*NSA funded project in RAISE Lab*

- Distributed a data privacy preserving algorithm (LACE) package in Python; tested the project through Travis-CI and provided documentations for the package. Applied my package to education and medical data sets. Evaluate data set utility through supervised learning.  
*Skills: Python, DevOps(Continuous Deployment), Data Deduplication, Data Cleaning*

### Building Movie Recommendation System

Aug 2015 - Dec 2015

*Course Project / "Netflix Prize" completion extension*

- Build a movies rating prediction and recommendation system by training from 100 million Netflix ratings by Factorization Machine, SVM and ANN. Accelerated the learning process through high performance computing (HPC) server.
- Crawled cast, critic reviews from rotten tomatoes and classified the movies basing on Jaro-Winkler Distance. Reduced the RMSE by up to 9% with the help of external information.  
*Skills: Scikit-learn, HPC, MPI, libFM*

### Continuous Integration/Delivery Pipeline

Aug 2016 - Dec 2016

*DevOps practice*

- Basing on abstract syntax tree, created a test-suite generator to *only* test the diff between two commits.
- Integrated Ansible scripts, Docker and Jenkins to build, test and deploy our "sunrise-calculator" application.  
*Skills: Continuous Deploy, AST, Ansible Script, Node.js*

## PUBLICATIONS

---

Chen, J., Nair, V., Krishna, R., and Menzies, T.. Is "Sampling" better than "Evolution" for Search-based Software Engineering. arXiv preprint arXiv:1608.07617 (2016).