Ravish Chawla
3900 RiverLook Parkway, Unit 104
Marietta, GA, 30067

Phone: (678) 332 – 6565
Email: rchawla@gatech.edu
Portfolio: www.ravishchawla.com
GitHub: www.github.com/ravishchawla

Education

Georgia Institute of Technology – Atlanta, GA

- Master of Science in Computer Science and Machine Learning (2015 – 2017)
  - GPA : 4.0 / 4.0 (overall)
- Bachelors of Science in Computer Science (2012 – 2014)
  - GPA: 3.83 / 4.0 (overall)
  - Graduated with Highest Honors

Work Experience

VMware AirWatch – Atlanta, GA

Research Engineering Intern (May – Aug 2016)

- Work on the Research Team to conduct research in, and develop applications for newer areas of research for the team.
- Applied knowledge of Machine Learning, and experience in the Android Platform to develop apps that used Online Learning to learn and predict behavior of users on device sensor and state data collected from the mobile devices.
- Filed 2 Patents for developing an approach to the problem of detecting and recognizing vehicle drivers, and enforcing usage policy in vehicles.

Zynga – San Francisco, CA

Software Engineer (Jan – May 2015)

- Worked on Cross Platform Game development using Unity Game Studio. Created and developed several services and features for the mobile game Zynga Poker as part of a team. Many of these features were released in updates and are currently being used by millions of users.
- Gained experience in working with the Unity Game Studio, C#, and Mono, and backend languages PHP and JavaScript.

Software Engineering Intern (May – Aug 2015)

- Worked on the Web Game Platform for Zynga Poker. Implemented utilities for developers and managers that allowed them to schedule push notifications for delivering in-game commodities to users of the platform.

Skills and Knowledge

- Programming in Java, C++, JavaScript, Python, C#, HTML/CSS, Objective-C, and PHP
- Experience in Android Development, Unity3D, Big Data, NOSQL and SQL.

Project Experience

- **SafeDriving Driver Detection and Recognition application for AirWatch Research**
  - Researched and implemented an application for detecting and recognizing vehicle drivers while they are using mobile devices. The application used Machine Learning to train on device sensor and GPS information, in order to build a model that can distinguish between drivers and passengers.
  - The application was built on Android, with a Python Server backend. In addition to the data collection, the app was responsible for data processing, model-based prediction, and enforcement on a high-frequency basis. The application was able to achieve higher than 95% accuracy in distinguishing drivers from passengers.
- **Social Hub for Zynga Poker**
  - Developed the Social Hub as part of a team for Zynga Poker. The Social Hub was designed to act as an in-game section that allowed users to communicate with other players in the game. Developed important services and features for the app, ranging from additions to the game backend to handle new calls to the Social Networking API, and work on the game components responsible for providing users the interface they could interact with.
- **Podium – A Multi-Attribute Ranking application for GT Visual Analytics Lab**
  - Developed a ranking system in JavaScript. The application inferred user-preferred attributes, based on how the user interacted with the rows in the app. As the user used the application, the ranking improved to align more with the user's preferred ranking preference, based on all the attributes affected by user's interaction with the app.
- **University Graduate Recommendation System**
  - Developed an application for recommending Graduate Schools to students. The application used Machine Learning models for prediction, trained on a dataset of the colleges students applied to, their education background, and acceptance into the colleges. The dataset for this application was scraped from forums, and consisted of over a hundred colleges in the U.S.
- **3D Foreign Object Detection on Augmented Displays in Computer Vision**
  - Developed an application for detecting foreign objects within a larger object. The application used an Intel RealSense ® Depth camera to separate out the object in a video frame, and computed the 3D difference between the object and its CAD model, and displaying the differences by augmenting them on the screen over the actual object.