Kevin Shimotakahara . Jarvis Consulting

I recently completed my graduate studies at the University of Ottawa, obtaining a Masters of Applied Science in Electrical and Computer Engineering. I qualified for $76,000 in scholarships and NSERC CREATE stipends during my two years at the University of Ottawa, including the nationally recognized NSERC CGS-M scholarship. My thesis topic was on improving the performance of device-to-device mobile communications via reinforcement learning algorithms to make them suitable for smart grid applications. My work has resulted in five publications, including two publications in journals created by the Institute of Electrical and Electronics Engineers (IEEE Access and IEEE Systems Journal).

I also have some industry data science experience, having developed electricity price forecasting neural networks with BluWave~AI, and a reinforcement learning-based electric vehicle charging controller for the world-renowned Fraunhofer Institute for Solar Energy Systems in Freiburg, Germany.

The work I enjoy most is solving problems with computers while applying the principles of discrete mathematics. As such, I am currently seeking rich opportunities to expand my skills in software development to a professional level; my time with Jarvis has significantly improved my programming skills, and I am eager to share them with the world.

Skills

Proficient: Java, Python, Matlab, Linux/Bash, RDBMS/SQL, Agile/Scrum, Git, Reinforcement Learning Algorithm Design, SpringBoot, Apache Maven, JDBC

Competent: JUnit/Mockito, RESTful APIs, Time Series Data Analysis, Machine Learning/TensorFlow, Overleaf (LaTeX)

Familiar: Google Cloud Platform, Microsoft Azure, Big Data Analytics, Hadoop/Hive/Zeppelin, Apache Spark, Azure Databricks, PySpark, Python/Flask web applications, C/C++

Jarvis Projects

Project source code: https://github.com/jarviscanada/jarvis_data_eng_KevinShimotakahara

Cluster Monitor [GitHub]: This project automates the monitoring of a Linux server cluster's activity. Machine usage data and hardware information gets stored in a PostgreSQL database, which is automatically populated by a network of server nodes every minute. The intended use case of this software is to help Linux Cluster Administration teams evaluate their server cluster's activity and compare it with their physical assets to inform their future assets planning decisions (e.g. adding/removing servers). The database host setup has been automated with shell scripts, and a .sql file has also been written that contains common queries used by system administrators. Moreover, we automatically configure the installation of PostgreSQL with a shell script that creates a Docker container containing PostgreSQL complete with a persistent Docker volume that stores the data managed by the PostgreSQL instance running in the container.

Core Java Apps [GitHub]:

- Twitter API App: This application allows a user to post, read, and delete Tweets from the Linux Command Line Interface. Our Dockerized Java program accepts a Twitter Developer's OAuth 1.0 credentials needed to be serviced by the Twitter REST API, in addition to the information about the post/show/delete Tweet action to be performed as command line inputs. The application then proceeds to contact the Twitter API to fulfill the request, and prints the JSON file(s) containing information about the Tweets that the user just posted/read/deleted. Apache Maven was used in this project for organizing its build workflow; part of this workflow includes unit/integration test phases, where Maven commands can execute rigorous software test scripts written with the JUnit test framework.
- JDBC App: This application sets up a connection via the JDBC API to a PostgreSQL database that manages a business' customer and order records. The core concept of this project was to develop objects that can streamline the process of building Strings that contain SQL statements which can be fed to JDBC APIs, whereupon the PostgreSQL driver that implements the JDBC interfaces can forward the SQL statements to the database application, and receive its response.
- Grep App: This application recreates partial functionality of the grep command commonly used in the Linux command line. More specifically, given an input regular expression, directory, and output path/filename, it will search through all files found in the directory and all its sub-directories for instances of the regular expression, and print the lines of text containing these instances into the indicated file. This app leverages Java streams to remain functional when its heap memory is smaller than the files it is reading.

SpringBoot Microservice Stock Trading App (REST API) [GitHub]: This project created a stock trading service in the form of a SpringBoot REST API application, where users could interact with a webserver to create an online account that they could deposit funds within, and then proceed to build and manage a securities portfolio. We provide up to date stock information from the online IEX stock exchange for our users to keep their investment strategies well informed. This web application uses SpringBoot to automatically configure Apache Tomcat, a Java web applet that can forward client requests to the components of our code that can handle them while converting the requests into a format our code can understand. SpringBoot was also used to facilitate constructing our application's interdependent objects and utilizing JDBC drivers to access our local database.

Python Data Analytics [GitHub]: This project analyzed an e-commerce store's customer transaction data using Python/Jupyter Notebook to process data stored in a PostgreSQL data warehouse. The sample distribution of purchase magnitudes was plotted and outlier data was managed for more meaningful sample statistics. Monthly revenue streams and customer activity were also analyzed visually. Finally, formal Recency, Frequency, Magnitude analysis was conducted on the store's clientele, classifying them into 11 different categories based on their behaviour. Three important categories were identified, and targeted marketing campaigns were suggested for these groups.

Hadoop [GitHub]: This project used Google Cloud Platform to provision a Hadoop cluster to store a CSV file containing World Development Indicator data in a Hadoop Distributed File System. Hive queries were designed and executed on the distributed dataset with the help of both Beeline and Zeppelin user interfaces to extract insights into the state of the world we live in. Query speeds were optimized through the use of Hive partitions and Parquet file formatting.

Spark [GitHub]: This project used Microsoft Azure Databricks to perform big data analytics workflows on retail transaction data that involved ingesting source data into a distributed file system; provisioning VM clusters; and running an Apache Spark Application as a distributed execution engine on top of a notebook user interface where PySpark code was written to run queries that answered key business questions. The queries performed in this project were the same as those performed in the aforementioned Python Data Analytics project, except PySpark DataFrames were used to characterize the queries instead of Pandas DataFrames.

Cloud/DevOps [GitHub]: Not Started

Highlighted Projects

Machine Learning-Based Electricity Price Forecasting: Designed, trained, and tested a regression neural network with Python/TensorFlow/Pandas to forecast the Hourly Ontario Electricity Price; neural network forecasts had a 24% lower mean absolute error relative to the Ontario Independent Electricity Systems Operator's prediction models over an entire year's worth of price data.

Reinforcement Learning Resource Allocation for LTE D2D-Enabled Smart Grid: Formulated a multi-agent Q-learning resource allocation algorithm aimed at minimizing the packet drop rates of smart grid devices in an out of network coverage scenario. The algorithm and engineering models were coded in Matlab/Simulink, leveraging Simscape Power Systems, SimEvents, and LTE System Toolbox. The algorithm outperformed the LTE standard algorithm, demonstrating 20%-40% absolute reductions in packet drop rate and 10-20 ms reductions in latency for all smart grid applications modeled. The results of this project have been published in two peer-reviewed IEEE journals. IEEE Access DOI: 10.1109/ACCESS.2019.2920662; IEEE Systems Journal DOI: 10.1109/JSYST.2020.3025202

Professional Experiences

Software Developer, Jarvis (2020-present): Solve classical software design and big data problems with professional-grade methodologies.

Scientific Assistant, Fraunhofer Institute for Solar Energy Systems (ISE) (January 2020 - May 2020): Fulfilled a four-month research contract aimed to integrate reinforcement learning electric vehicle charging controller logic into Fraunhofer ISE's Multi-Agent Energy Management System project. Played a developer role in a GitFlow software development environment, interfacing novel Python scripts with Dockerized REST APIs. Produced simulation results suggesting novel approach could make the microgrid's utilization of local photovoltaic energy generation 16.7% higher than that of conventional charging schemes. The results of this project were showcased at the DACH+ Energy Informatics 2020 conference held in Switzerland.

Assets Management Engineering Co-op, Hydro Ottawa Ltd. (May 2016 - December 2016): Designed a database application in Access VBA to determine HOL's distribution system's capacity for connecting distributed energy resources (DER) of all kinds including rooftop photovoltaics. Developed a sustainable information system to power application that accesses regularly updated databases that characterize the nearly 6,000 km of conductors, etc. that constitutes power system. The application automated several steps in DER connectivity impact assessments, resulting in a significant reduction of engineering labor costs associated with connecting renewables to the power grid. Succeeded in laying out the groundwork for a means to collect information from heterogeneous structured/semi-structured/unstructured data sources, perform business logic on the data, and present it to the end-user via a Microsoft Access front end.

Research Assistant, Defense Research and Development Canada (May 2015 - August 2015): Developed ray-tracing programs in Matlab capable of simulating the propagation path of 9-26 MHz radar beams as they refract through the ionosphere. Succeeded in obtaining publishable results in a four-month period as a second-year undergraduate co-op student. DOI: 10.1109/IRS.2016.7497370

Education

University of Ottawa (2018-2020), Master of Applied Science (MASc), Electrical and Computer Engineering - NSERC CGS-M Scholarship - NSERC CREATE TOPSET Stipend - NSERC MSFSS Scholarship - University of Ottawa International Experience Scholarship - University of Ottawa Ontario General Scholarship - University of Ottawa Excellence Scholarship - University of Ottawa Entrance Scholarship - GPA: 4.0/4.0

Carleton University (2013-2018), Bachelor of Engineering, Electronics - NSERC USRA - Carleton Entrance Scholarship - Dean's List 2013-2018 - 1st Place: Carleton Junior Design Competition 2014 - GPA: 4.0/4.0

Miscellaneous

- University of Alberta Reinforcement Learning Specialization
- Hockey player
- Musician - guitar, bass, vocals, songwriting
- Long-distance runner, cyclist
- Strategy game enthusiast
- Home cook, design cost and flavour optimized recipes with sustainable ingredients