Sirshendu Ganguly

sganguly@wpi.edu | (774) 701-8642 | Worcester, MA | linkedin.com/in/sirshendu-ganguly | github.com/Sirsho1997

Education

MS - Data Science — Aug 2021 - Present
Worcester Polytechnic Institute; GPA: 3.5/4.0 — Worcester, MA
Courses: Algorithms, Introduction To DS, Business Applications in ML, Statistical Methods For DS

M.Sc - Computer Science — Jul 2018 - Oct 2020
St. Xavier's College; CGPA: 7.6/10 — Kolkata, India
Activities: Class Representative, and Editorial Board Member

B.Sc - Computer Science — Jul 2015 - Jun 2018
University of Calcutta; CGPA: 7.6/10 — Kolkata, India
Passed with First-Class in Honours

Skills Summary

Programming Languages: Python (Advanced), C/ C++ (Intermediate), Java (Intermediate), R (Beginner)
Frameworks: NumPy, Pandas, Matplotlib, Seaborn, Scikit-Learn, OpenCV, NLTK, Keras, TensorFlow, PyTorch
Cloud AutoML services: Azure ML, DataRobot
Query Languages: SQL with experience in OracleSQL, NoSQL with experience in MongoDB
Tools: Git, LaTeX, and Docker
Certification: Deep Learning Specialization on Coursera, taught by Andrew Ng

Work Experience

Research Assistant — Aug 2021 - Present
Worcester Polytechnic Institute — Worcester, MA

In Dr. Lorenzo De Carli's lab

- Designed a line of defense *against modern-day ransomware attacks*, by leveraging file-system activity; developed a *low-level I/O packet sniffer* that monitors the machine's file system activities such as the number of bytes read per second, written per second, and change in file entropy; used state-of-the-art machine learning models to *detect the presence of any ransomware* using the data retrieved by the I/O sniffer.
- Proposed a novel approach of moving-target defense to *counter security issues in application-level network protocol used in embedded/IoT space*; arising due to software homogeneity and thus threatening large installed base e.g. power grid; *injected syntax mutation* at the granularity of individual deployment, implemented via a set of heuristics based on the candidate program data structures and functions via static transformations; the mutations were dynamically tested for functional correctness.

In Dr. Fabricio Murai's lab

- Currently conducting experiments using various *Graph Neural Networks* for detecting *fraudulent transactions.*

Assistant System Engineer — Jan 2021 - Aug 2021
Tata Consultancy Services — Kolkata, India

- Maintained European and Asian country's servers for an US-based Bank.
- Daily activities required me to write python scripts for operating system-dependent functionalities, generate graphs depicting daily changes in the volume of data received, and derive insights.

Computer Science online instructor (part-time) — Aug 2020 - Dec 2020
MyCareerwise — Kolkata, India

- Advised by Dr. Debashis Roy, founder of MyCareerwise. (https://mycareerwise.com/)
- Curated content useful for cracking technical job interviews, and entrance exams such as GATE, NET, etc.

Projects

Distinguishing the Monkeypox cases from the Non-Monkeypox cases using lesion images
Computer Vision

- Made use of the *Monkeypox Skin Lesion Dataset* hosted on Kaggle; the original dataset contains a total of 4251 images (Monkey Pox: 1831, Others: 2420) which also includes augmented images generated by using methods such as rotation, translation, reflection, shear, hue, saturation, contrast, brightness jitter, noise, scaling, etc.
- Implemented a *ResNet9* architecture using PyTorch from scratch and used PyTorch's own implementation of pre-trained *Resnet34* for the given classification task. Both of the networks implemented used the Weight Decay Regularization technique and the Adam Optimizer; accuracy is used to compare the model's performance.
- ResNet9 provided the best accuracy of *86%*, for a train:val:test ratio 80:10:10, and ResNet34 provided the best accuracy of *97%*.
- Python libraries used: PyTorch, and Matplotlib.

https://github.com/Sirsho1997/MonkeyPoxDetectionUsingCNN

Forecasting opening stock price by using LSTM
Recurrent Neural Network

- Crawled *Netflix stock dataset* from Yahoo finance containing stock data (opening, closing,high, low, adjusted closing) for the past 10 years (5082 rows, and 5 columns); For simplicity, only one variable i.e. the "open" price is used for analysis.
- Implemented an *LSTM architecture*, with MSE as the loss function, and the RMSProp optimizer to find the optimal weights, to run on a train-test split of 80:20, where the features contain prices appended from the last 75 days, and the price for the next day is chosen as the dependent variable.
- A RMSE of *33.97* is achieved, which is ideal in this case because as we know that stock price value depends on multiple variables such as a change in top management, seasonal inflation rates, etc for which data is hard to collect.
- Python libraries used: Keras, Pandas, NumPy, and Matplotlib.

https://github.com/Sirsho1997/NetflixStockPricePrediction

Multiclass Node Classification using Graph Convolutional Network
Graph Neural Network

- Utilized the *AmazonCoBuy dataset* which contains 13,752 nodes(divided into 10 classes) and 4,91,722 edges. Nodes represent different goods bought via Amazon and edges represent that the two goods are frequently bought together. Each node has 767 features which are the bag-of-words embedding of the product reviews.
- Implemented the *Graph Convolutional Network (GCN)* model with 2 layers and trained it using the Adam optimizer with cross-entropy selected as the loss function.
- Achieved best validation accuracy of *87.3%*.
- Python libraries used: DGL, PyTorch, Matplotlib, and Seaborn.

https://github.com/Sirsho1997/GraphNodeClassificationUsingGCN

Determining the genre of a book by parsing book summary using NLP
Natural Language Processing

- Worked on the *CMU Book Summary dataset* containing book summaries for 16,559 books.
- Performed preprocessing tasks such as *Stemming*, *Lemmatization*, and *Stop Words Removal* on book summary and then used *tf-idf statistic* to extract vectors representing each word's importance.
- Analyzed the dataset using basic different machine learning models such as *LR*, *Linear SVM*, and *Non-Linear SVM*, and achieved an accuracy of *79.55%* using *Non-Linear SVM* on a train-to-test split of 85:15.
- Python libraries used: Pandas, NumPy, NLTK, Scikit-Learn, Matplotlib, and Seaborn.

https://github.com/Sirsho1997/Book-Genre-Prediction-using-Book-Summary

Report: https://github.com/Sirsho1997/Book-Genre-Prediction-using-Book-Summary/blob/master/Report.pdf

Predicting network switching among telecom prepaid customers
Business Analytic

- Utilized a *private dataset provided by a Telecom Operator* which contained 75 features such as total call count in last month, last recharge date, number of messages sent, etc and 1,57,747 records; the dataset contained 14% of users who switched networks.
- Performed preprocessing tasks such as *missing data imputation using linear regression*, *data aggregation for removing multicollinearity*. Analyzed the dataset by using various classification models (*LR, SVM, KNN, Decision Tree*, and *Random Forest*), by first balancing the dataset using *SMOTE* undersampling and oversampling algorithm.
- The Random Forest algorithm provided the best accuracy of *92%* for a train-to-test ratio of 70:30.
- Python libraries used: Pandas, NumPy, Scikit-Learn, Matplotlib, and Seaborn.

https://github.com/Sirsho1997/Telecom_Customer_Churn_Prediction

Program analysis for understanding and reverse engineering of network protocols
Comparison between the various state-of-the-art reverse engineering tools

- Analyzed the current state of *reverse engineering* in the domain of network engineering and evaluated several research papers on the topic.
- We first started with the manual reverse engineering techniques which soon were replaced with automatic implementations.
- We also briefly discussed the difference between network-traffic based automatic protocol reverse engineering techniques and application-level-based automatic reverse engineering techniques.

Report: https://github.com/Sirsho1997/Monograph-of-Program-Analysis-for-Understanding-and-Reverse-Engineering-of-Network-Protocols/tree/main

Publication

T. Ren, R. Williams, S. Ganguly, L. Lu, L. De Carli. **Breaking Embedded Software Homogeneity with Protocol Mutations.** EAI SecureComm 2022.*(Conference)*

PDF: https://ldklab.github.io/assets/papers/securecomm22-mutations.pdf